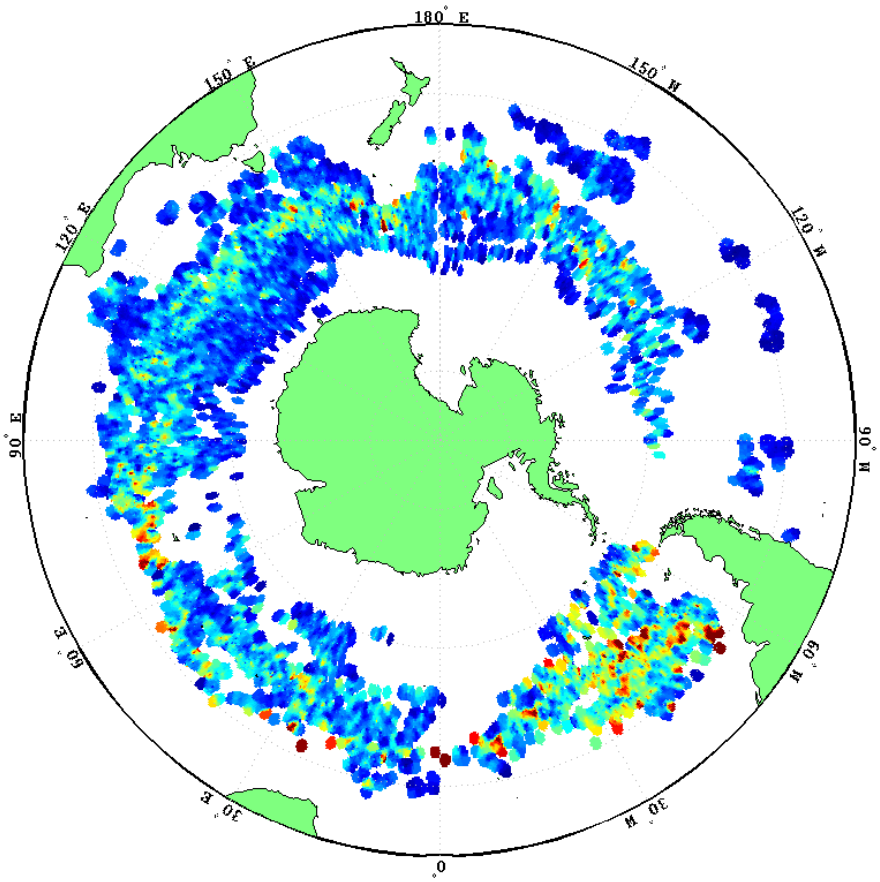
Validation of the SatGEM Velocity Dataset using Argo Data



by

Christopher Roach

Bachelor of Science,
University of Tasmania

A thesis submitted in partial fulfilment of the requirements of the Bachelor of Antarctic Studies with Honours
Institute of Antarctic and Southern Ocean Studies (IASOS)
University of Tasmania
June, 2009

Declaration

I hereby declare that the thesis contains no material which has been accepted for the award of any other degree or diploma in any tertiary institution and that to the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made in the text of the thesis.

Christopher Roach

June 2009

Abstract

The 4D SatGEM velocity model of the Southern Ocean has been derived from satellite altimetry and a Gravest Empirical Mode (GEM) model of temperature and salinity. The validity of the SatGEM model is assessed in this study by numerical forward modelling of Argo float trajectories through the 4D velocity fields.

Univariate statistical analysis of simulated and actual path lengths, together with analysis of the misfit between simulated and actual surfacing positions, suggests that the SatGEM model provides an adequate representation of the gross features of the Southern Ocean. However, on a local scale there is significant variability in misfit due to a combination of statistical fluctuations, the limited SatGEM model resolution and complex local velocity fields.

A high proportion of large relative misfit cycles are apparent for short Argo path lengths suggesting that the Argo float trajectories in these areas may be strongly influenced by complex small-scale flow features that cannot be adequately resolved in the SatGEM model. Argo float cycles with path lengths in excess of 50km and small relative misfits are effectively randomly distributed. However statistically significant clusters of large relative misfit cycles are apparent particularly in areas of known small-scale eddy current development such as east of the Kerguelen Plateau and also in the Brazil-Falklands Convergence where GEM models based on satellite altimetry are known to be inaccurate. The angular misfit for medium and small relative misfit Argo cycles is normally distributed suggesting that variability in this case arises due to statistical processes. The uniform distribution of angular misfit for large relative misfit cycles is suggestive of more random chaotic processes likely related to flow bifurcations in zones with complex velocity structure. In these cases the relative misfit is highly sensitive to the initial descent position of the float in the velocity model.

Validation of the SatGEM model using Argo float data suggests that velocity models derived from satellite altimetry and GEM derived temperature and salinity fields can provide adequate representations of ocean dynamics for large-scale investigations but the small-scale applications of these models are limited primarily by their spatial resolution and hence their inability to accurately model complex local flow features.

Acknowledgments

I would like to acknowledge and thank the following people:

My supervisors, Professor Nathan Bindoff and Dr Helen Philips for the guidance and assistance they provided over the course of my honours project. Nathan, thanks for finding the time out of your busy schedule over the last month to help with my interpretation of results. Helen, thanks for reading all the drafts I sent you, I hope you didn't mind my all too frequent emails clogging your in box.

Andrew Meijers for creating the SatGEM fields; providing advice on navigating the dataset and giving me access to the sea surface height fields used in SatGEM creation.

Thanks to my dad, Michael, for all the late nights spent proof reading my numerous rough drafts and many helpful suggestions during the analysis.

Dr Julia Jabour, Dr Kelvin Micheal and the other staff of the Institute for Antarctic and Southern Ocean Studies and the Antarctic Climate and Ecosystems CRC for the support provided during the coursework.

To my fellow IASOS honours students, Hamish and Nick and to all the masters' students I shared the office with.

I would also like to acknowledge Phil Morgan, Lindsay Pender and the other CSIRO staff who developed and made publically available the SEAWATER toolkit for Matlab (http://www.cmar.csiro.au/datacentre/ext_docs/seawater.htm).

The data used in this study were collected and made freely available by the International Argo Project and the national programs that contribute to it. (http://www.argo.ucsd.edu, http://argo.jcommops.org). Argo is a pilot program of the Global Ocean Observing System.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Background

Over the past two decades a number of significant advances have occurred in the field of physical oceanography. Major programs like the World Ocean Circulation Experiment (WOCE) and the Argo program have dramatically increased the resolution of the hydrographic coverage of the world's oceans. The coverage and quality of satellite altimetry data has improved substantially and new analysis techniques have been developed.

The Gravest Empirical Mode (GEM) method is an example of a new analysis technique which seeks "low dimensional projections of the leading order dynamics of the ocean" (Sun and Watts 2001). GEM techniques have been combined with the use of inverted echo sounders or satellite altimetry to create reconstructions of the sub-surface temperature, density and salinity structure of regions of the ocean with a high degree of accuracy (Meinen and Watts 2000; Sun and Watts 2001; Watts et al. 2001; Mitchell et al. 2004; Nardelli and Santoleri 2005). With the extension of GEM techniques from 2D to 3D datasets (Sun and Watts 2001; Meijers et al. 2009b) and improvements in resolution it has become possible to use the resulting temperature, salinity and density fields to produce density driven (geostrophic) velocity fields. These velocities by definition lie parallel to streamlines and proxies for streamlines such as contours of sea surface height (SSH).

1.2 Aims and Objectives

While GEM-derived geostrophic velocities have been used to estimate transport since Meinen and Watts (2000) first applied the GEM-techniques to the North Atlantic the creation of high resolution, temporally varying 3D velocity fields is a recent development (Meijers et al. 2009b, 2009a). These new velocity fields for the Southern Ocean (SatGEM fields) have undergone only limited tests and validation. The principal aim of this study is to attempt to further validate the SatGEM velocity fields by comparison with velocity information derived from Argo float data.

Techniques used to develop the SatGEM velocity fields are designed to estimate geostrophic velocities only, and while methods exist to introduce Ekman velocities (near surface wind stress driven velocities) to the near surface layers other components of flow and transport such as boundary currents, inertial oscillations and

mixing may either not be well resolved or else missed entirely by the SatGEM fields. The temporal and spatial scale of the SatGEM fields also means that features with duration of less than 7 days or smaller than 1/3° cannot be resolved. Features up to twice these dimensions may be only poorly resolved.

In contrast to the SatGEM fields subsurface drifters such as Argo floats are sensitive to the contribution of all ocean current systems. As a result in regions in which flow components not resolved by the SatGEM process are significant a degree of divergence between Argo data and the SatGEM fields is to be expected. The second major aim of this project is identify and investigate major discrepancies between the two datasets as a means to identify regions where processes other than geostrophic flow may make a significant contribution to the overall velocities. Identification of regions of non-geostrophic flow is important in understand processes such as the Meridional Overturning Circulation which is a major element of the global climate system.

A final goal of this study is to examine relationships between external variables such as sea surface height, depth and time on the performance of the SatGEM velocity model.

1.3 Methodology

While there are a number of independent data sources against which the SatGEM velocity fields can be tested, most were not suitable: current meters provide long term velocity data but only over very limited areas; ship-based measurements are costly and sparse in both geographic and temporal distribution while surface drifters are too heavily influenced by factors not properly modelled by GEM based techniques. This left Argo floats as the most plausible dataset. Despite the use of PALACE floats to generate and validate velocity fields during WOCE there has been little use of Argo data for this purpose. While techniques have been developed to estimate velocities at depth from profiling float surface fixes (Park et al. 2005; Lebedev et al. 2007), these produce mean velocities for an entire Argo cycle or a point estimate at the mid-point of the cycle, and as so the utility of such methods for the validation of the SatGEM fields are relatively limited. As a result it was decided to approach the issue from a numerical modelling perspective by designing and implementing a model capable of simulating the trajectory of an Argo float moving through the SatGEM velocity fields. Path lengths between Argo float descent point and both simulated and real surfacing

points were determined and the misfits between simulated and real surfacing points were calculated to provide a numerical measure of the local spatial and temporal validity of the SatGEM velocity fields.

Chapter 2: Literature Review

2.1 The Southern Ocean

Currents and Fronts

A number of currents play a role in the dynamics of the Southern Ocean. On the northern fringes of the Southern Ocean the subtropical gyres dominate. Further south lies the Antarctic Circumpolar Current (ACC) and adjacent to the Antarctic land mass itself gyres occur in the Weddell and Ross Seas. A map of these features is shown in Figure 2.1.

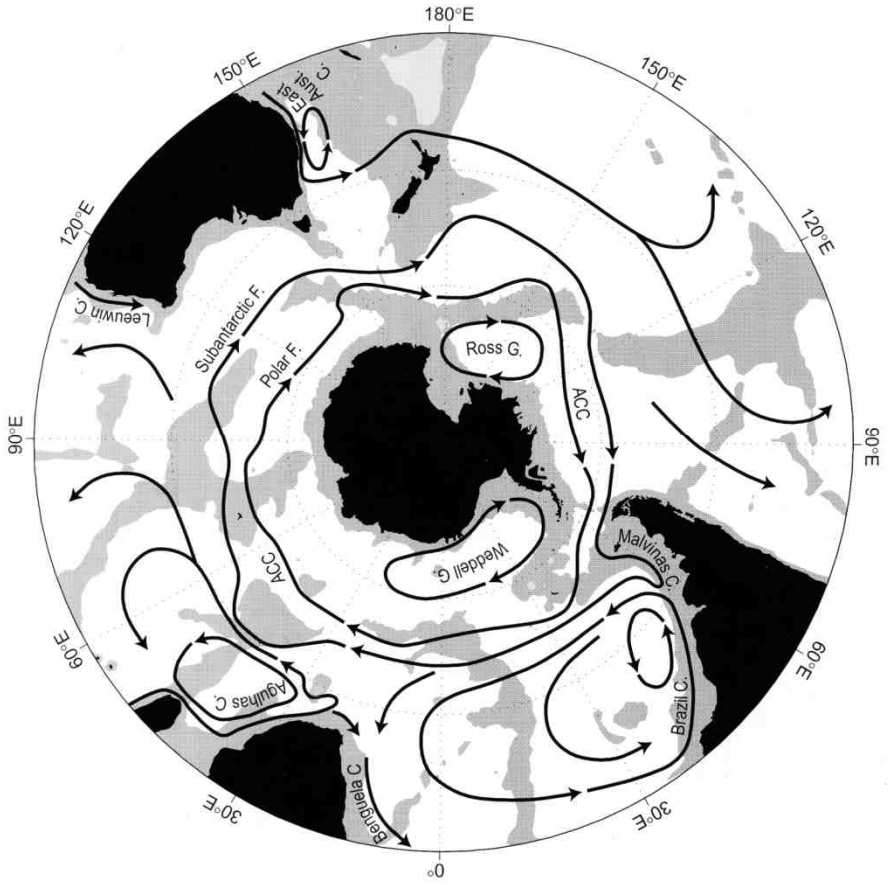


Figure 2.1*: Major currents and other features of the Southern Ocean. From Rintoul et al. (2001).*

Of all these currents the ACC is the most significant carrying an average of between 97 x$10^6$$m^3$/s (Orsi et al. 1995) and 134 x$10^6$$m^3$/s (Rintoul et al. 2001) of

water around the globe and forming a vital link between the ocean basins. The ACC is deep reaching and as a result its path is heavily influenced by the bottom topography.

The boundaries between the various regions of the Southern Ocean are marked by sharp transitions between water masses known as fronts (Stewart 1997). Generally, four major circumpolar or near circumpolar fronts (Figure 2.2) are recognized (Orsi et al. 1995): the Subtropical Front (STF, blocked by South America); the Subantarctic Front (SAF); the Polar Front (PF) and the Southern ACC Front (sACCf or SACC). The criteria used to define these fronts have varied considerably from study to study, for examples see Belkin and Gordon (1996). This has resulted in discrepancies in the locations of fronts (Sokolov and Rintoul 2007). In addition to the circumpolar fronts two minor fronts (the Scotia front and an unnamed front) separate the Weddell and Ross gyres from the ACC.

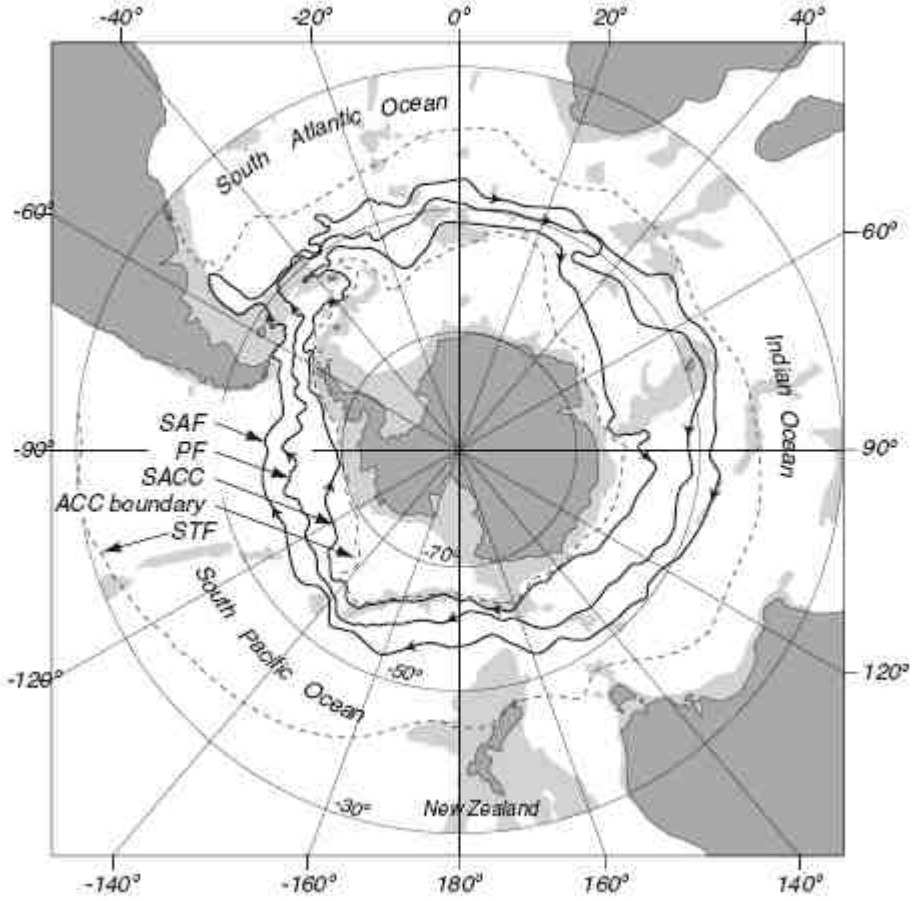


Figure 2.2*: Schematic map of the principal fronts of the Southern Ocean. From Stewart (1997), after Orsi et al. (1995). Shaded areas indicate depths of 3000m or less.*

Based on studies in the Drake Passage it was long believed that transport along the ACC was concentrated in a number of continuous and deep jets associated with the circumpolar fronts (Rintoul et al. 2001). The advent of high resolution models; remote sensing of the jets and higher resolution observations have since shown that the ACC is considerably more complex and is actually composed of (Hughes and Ash 2001): "a complex interweaving of jets, breaking and joining, beginning and ending". A number of these jets are associated with the temperature and salinity features which older studies have used to define frontal location and are thought to account for some of the discrepancies in frontal positions in those studies. Similar studies (Sokolov and Rintoul 2007) also indicate that frontal structure is also more complex than previously believed.

The Overturning Circulation and Water Masses

While the circumpolar transport of the ACC is the dominant feature of the Southern Ocean the weaker meridional overturning circulation plays a very significant role in global climate. The circulation is dominated by six major water masses, the names and properties of which are shown in Table 2.1.

Water Mass	Temperature (°C)	Salinity (psu)	Other Characteristics	Sources
Subantarctic Mode Water (SAMW)	4-15	34.2-35.8		(Hanawa and Talley 2001)
Antarctic Intermediate Water (AAIW)	3-5 (Potential temperature)	34.2-34.4		(Gordon 2001)
Upper Circumpolar Deep Water (UCDW)	1-2 (Potential temperature)	34.2-34.4	Oxygen minimum	(Gordon 2001) (Orsi et al. 1995)
North Atlantic Deep Water (NADW)	4	35		(Stewart 1997)
Lower Circumpolar Deep Water (LCDW)	1-2 (Potential temperature)	34.2-34.4	Salinity maximum	(Gordon 2001) (Orsi et al. 1995)
Antarctic Bottom Water (AABW)	<-1 (Potential temperature)	34.65-34.75		(Gordon 2001)

Table 2.1: *Defining Features of Southern Ocean water masses.*

The densest of these water masses is the cold and saline AABW which originates on the continental margins of Antarctica, most notably in the Weddell Sea. Further up the

water column sits the LCDW, NADW and UCDW, south of the Polar Front both these water bodies extend to the surface. North of the Polar Front the AAIW and subsequently (north of the SAF) the SAMW dominate the near surface layers of the Southern Ocean.

The overturning circulation (Figure 2.3) is largely dependent on the balance between the southwards flow of the deep waters and the northward flow of AABW and surface waters north of the PF (Rintoul et al. 2001). It has been surmised that this process is at least partially driven by Ekman (wind driven) transport: Over much of the Southern Ocean westerly winds dominate (causing northward Ekman transport), reaching a maximum in the vicinity of 50°S; further south winds are weaker and more variable. This results in divergent Ekman transport which causes upwelling of the deep waters.

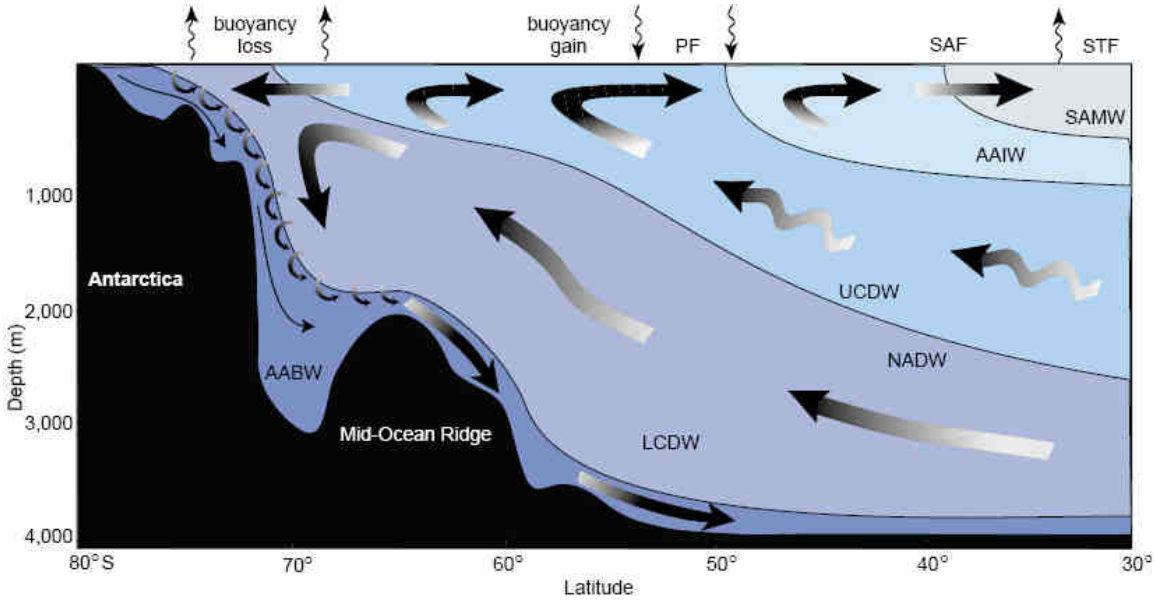


Figure 2.3: *Schematic diagram of the overturning circulation. From Speer et al. (2000).*

As a result of the circumpolar nature of the ACC meridional geostrophic transport is confined to regions in which sea floor topography cuts across the ACC. From this it has been deduced (de Szoeke and Levine 1981) that in order to balance northward Ekman flow and heat loss to the atmosphere eddies must contribute to an unusually large portion of the heat transport across the ACC. Models (Speer et al. 2000) and experiments (Phillips and Rintoul 2000) seem to confirm this. Areas associated with high eddy intensity include the Kerguelen and Campbell Plateaus.

2.2 The Argo Program

One of the major problems for Oceanographers has been the limitations on quality, quantity, geographic and temporal coverage of data. The development of the expendable bathythermograph (XBT) in the 1960s and the World Ocean Circulation Experiment in the 1990s (Roemmich et al. 1998) have done much to improve the extent and quality of coverage, however many limitations still remain. XBTs are principally deployed from ships of opportunity, and as such, the resulting data is mainly restricted to major shipping lanes and the measurements only provide temperature profiles. The WOCE survey provided high quality data across the entire water column, but even given seven years of work the spatial coverage remained sparse. A solution to some of these problems has been developed in the form of the Argo program.

The Argo program is a multi-national project to deploy and maintain an array of profiling floats to give global coverage of the upper ocean (Roemmich et al. 2001). The array consists of over 3000 floats spread throughout the world's oceans, each capable of taking temperature and salinity profiles to depths of around 2000m. The array produces over 100,000 profiles per year (Park et al. 2005). As part of the requirements of the Argo program these temperature, salinity and trajectory data are made available to the public via a number of data centres. These data are available in both near real time and delayed time, the latter with more stringent quality controls.

History

The Argo program is the latest in a long line of oceanographic experiments to make use of sub-surface floats. The use of sub-surface floats was pioneered by John Swallow in 1955 (Gould 2002) when he produced the first SOFAR (SOund Fixing And Ranging) floats. These floats consisted of a simple length of aluminium tubing with an electronic circuit and batteries sealed within the tubing and a transducer hanging below, producing a periodic acoustic signal. By triangulating the signal from multiple receivers it was possible to establish the float's location.

The first floats had an operational frequency in the region of 10kHz, which resulted in a detection range of around 5 km (Warren and Wunsch 1981). Later changes to the signal frequency allowed improvements in the detection range of these floats, and by the mid-1970s it had become common to use the SOFAR channel (which operates at frequencies near 1000Hz) which allowed floats to be detected over

ranges of up to 700km (Warren and Wunsch 1981). During the 1980s RAFOS (reverse SOFAR) floats were developed (Gould 2002). Like the SOFAR floats the RAFOS system relied on acoustic techniques to locate the float. However, unlike the SOFAR system RAFOS had the receivers fixed to the float and the signals transmitted from fixed stations. The float would then store the location data generated by triangulating the sound sources until a pre-set time when it would surface and transmit the data to a satellite.

With the advent of the WOCE program during the 1990s RAFOS floats were supplemented by the introduction of Autonomous Lagrangian Circulation Explorer (ALACE) floats (Gould 2002). ALACE floats could be set to maintain a desired depth for a period of time before surfacing then descending and repeating the process. Unlike the earlier generations of floats, ALACE floats did not make use of acoustic tracking; data on the location of a given float was obtained solely from fixes obtained by satellite when the float periodically surfaced. This had the downside of losing some of the details of movements in between the surfacing times. Towards the end of WOCE a number of ALACE floats were equipped with sensors for obtaining profiles of temperature and/or salinity while surfacing; producing Profiling ALACE floats (PALACE).

The Argo project itself is in many ways the successor to the hydrographic element of WOCE (Roemmich et al. 1998). The development of the Argo array was first proposed in 1998 in conjunction with the Global Ocean Data Assimilation Experiment (GODAE) and the Climate Variability and Predictability Program (CLIVAR). The first floats were deployed in 2000 and worldwide coverage being achieved by 2004 (Lebedev et al. 2007). By the end of 2007 the initial target of 3000 active floats had been met.

Argo Floats

With the exception of some limited use of fixed profiling devices in the Arctic the vast majority of data collected by the Argo project is sourced from profiling floats. Currently the vast majority of deployed Argo floats are of three main models: APEX, SOLO and Provor. All of these float models have similar characteristics: a nominal life time of 4 years or between 150 and 170 cycles and a maximum operating depth of about 2000m. All Argo floats use an external hydraulically operated bladder to control depth. By adjusting the volume of the bladder the volume and hence density of the float can be changed, allowing the float to rise or sink until neutral buoyancy is achieved.

An Argo float cycle (Figure 2.4) consists of four distinct phases. First is the surface phase during which data is transmitted to satellites.

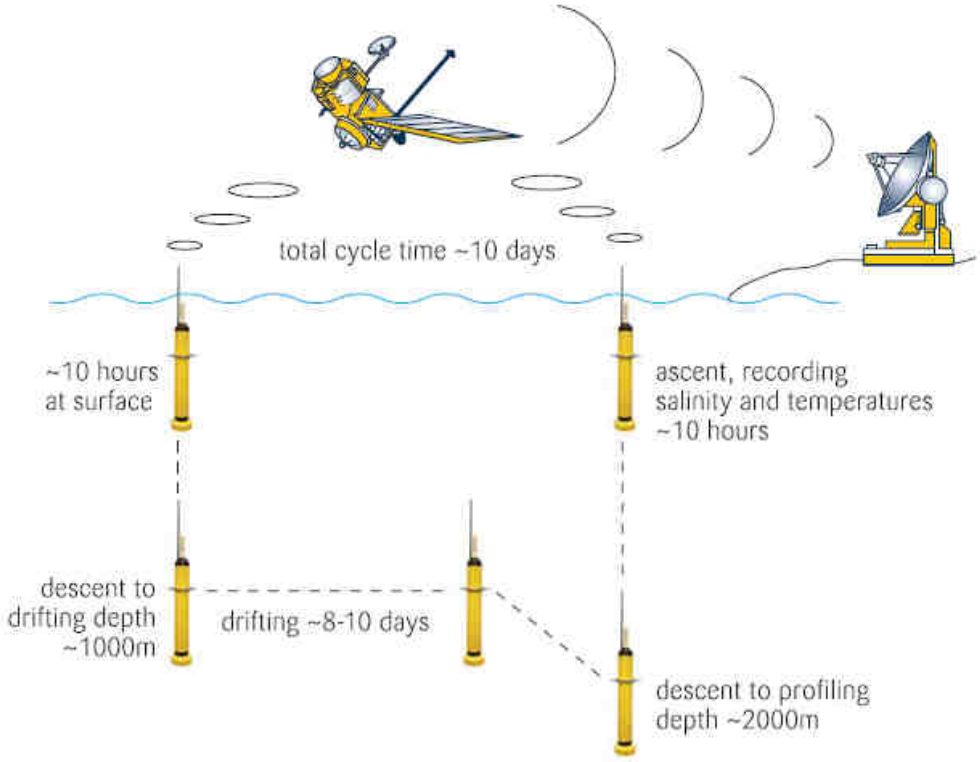


Figure 2.4: *Outline of a typical Argo float cycle. From Gould (2006).*

This is followed by descent to parking depth and an extended period spent at a parking depth which for 90% of floats is between 1000m and 2000m (Lebedev et al. 2007). Finally, the profiling phase in which temperature and salinity sensors record data while the float ascends to the surface. Dependent on the intended use of a float the profile may commence at the parking depth or may be preceded by a descent to a

greater depth. On average an Argo float cycle lasts for 9.5 days, of which 9 hours are spent on the surface (Lebedev et al. 2007).

Despite the use of the PALACE floats to estimate velocities of the deep Ocean during WOCE the use of Argo floats for a similar purpose has been limited.

2.3 Estimating Sub-Surface Structure From Surface Measurements

There are currently several major methods by which surface measurements can be used to model the sub-structure of a given region of ocean. Most older methods are based around the use of Single Empirical Orthogonal Function (sEOF) decomposition of hydrological profiles to indentify a correlation between the sEOF modes and a surface measurement (Nardelli and Santoleri 2005). More recently developed methods are based around the use of the Gravest Empirical Mode (GEM) technique, in which a 'unique' profile is associated with specific values of a function of the integrated density anomaly of a column of water.

The use of sEOFs to reconstruct a temperature or salinity profile from surface data originated in the late 1980s (Carnes et al. 1994). A typical method used for sEOF reconstructions was described by Pascual and Gomis (2002):

A set of profiles located at (x,y) with data recorded at a number (N) of common pressure levels is denoted by $\alpha_{x,y}(p)$. By subtracting the mean historical profile from each profile in $\alpha_{x,y}(p)$ a set of anomaly profiles, $\alpha'_{x,y}(p)$, can be created. From the profiles in $\alpha'_{x,y}(p)$ a NxN matrix can be constructed and rearranged so as to obtain a set of orthogonal eigenvectors (the EOFs) and the ratio of a given eigenvector to the sum of the eigenvectors will yield the fraction of variance explained by that particular EOF. Profiles can then be reconstructed from the equation:

$$\alpha_{estimated\ \ x,y}(p) = \sum_1^i A_i(x,y) EOF_i(p)$$

Where A_1 to A_i are the magnitudes of the corresponding EOFs.

Thus, if the amplitudes of each EOF can be associated with a surface property a profile of a given variable can be reconstructed. In practice it is usually not necessary to use all EOFs as a reduced number will usually account for the majority of the variance.

A gravest empirical mode technique is a method by which (Meijers et al. 2009a) the structure of variables such as temperature and salinity in a water column are parameterised as a function of integrated water column density or proxies such as

acoustic travel time or dynamic height. This approach allows the creation of a look up table of profiles of the variables; with each profile corresponding to a particular value of the function of density. In theory a given value of the function of density could correspond with a vast number of different temperature (T) and salinity (S) profiles. However the GEM technique works because it has been established empirically (Meinen and Watts 2000) that for some regions of the ocean each value of the density function is associated with a set of very similar profiles.

In addition to the GEM and sEOF methods two other techniques, coupled pattern reconstruction (CPR) and multivariate EOF reconstructions (mEOF), have been introduced in recent years. While initial work in the North Pacific (Nardelli and Santoleri 2005) suggests both these methods may result in greater accuracy than sEOFs and GEMs there has not been sufficient studies to establish the utility of mEOFs and CPRs more generally.

Comparison of GEM and sEOF Techniques:

There have been few direct comparisons of the 'real world' effectiveness of GEM and sEOF techniques. Sun & Watts (2001) applied a 2D EOF analysis and a geopotential height GEM to a number of transects in the Southern Ocean. While details of the EOF are not fully discussed in the paper, it is stated (Watts et al. 2001): "the resulting first mode of the EOF analysis was the same (for practical purposes) as the GEM fields".

Nardelli and Santoleri (2005) conducted a comparison of sEOF, GEM, CPR and mEOF models' performance for the North Pacific given varying "training times" (length of data availability). The two GEMs used displayed a similar level of error to existing climatologies for both salinity and temperature and showed little improvement with extension of the training period. GEMs also displayed little sensitivity to errors in sea surface temperature (SST) and salinity (SSS) measurements. The performance of the two different sEOFs (one based on SST and height and the second based on SSS and dynamic height) yielded less consistent results: The SSH and SST based EOF generally displayed a lower level of error in temperature estimates than the GEMs, while salinity estimates varied from slightly worse to slightly better than the GEMs as training period increased. The SSH and SSS based sEOF models displayed a significantly greater degree of error than the GEMs for short training periods but at training periods of 8 years or more the sEOF models displayed less error than the GEMs. Both sEOFs displayed greater sensitivity in surface errors

than the GEMs. Despite the apparently superior performance of sEOFs, Nardelli & Santoleri (2005) noted that: "sEOF-R methods are more likely to produce completely wrong predictions if they are trained with datasets that are not optimal".

2.4 Development Of The Gravest Empirical Mode Technique:

A GEM Of The North Atlantic Current:

The first GEM (Meinen and Watts 2000) was applied to a transect across the North Atlantic Current (NAC) stretching from 43°N 49°W to 42°N 44°W. The experiment made use of four current meter moorings and six PIES (inverted echo sounders fitted with pressure sensors) spread along an oceanographic transect in addition to 191 CTD casts in the surrounding region. This particular GEM used the round trip acoustic travel time (τ) of a 10 kHz pulse emitted from a PIES as the parameterised variable.

Data preparation involved correcting the τ time series for seasonal variations in the thermocline and then calibrating the seasonally corrected τ time series to equivalent time series for a fixed pressure level of 2000 dbars. Additionally, a number of CTD profiles were used to simulate τ (denoted as τ_{sim}) for each profile. At a fixed pressure level a cubic smoothing spline was applied to T as a function of τ_{sim}. Values of T were then calculated for a regular grid of τ from the spline. This procedure was repeated on other pressure levels at an interval of 25 dbars up to a limit of 5000 dbars. This resulted in the creation of a regular grid of T as a function of τ and pressure. A similar method was also employed to generate profiles of the specific volume anomaly (δ).

In order to test the usefulness of the GEM method Meinen and Watts (2000) used two approaches; estimation of T sections from return times using the GEM, and comparison of T estimates at individual PIES with data from nearby moored instruments.

By using the grids of T and δ as functions of τ and p as look up tables for simulated τ values based on CTD casts taken on two voyages in 1993 and 1994 Meinen and Watts attempted to reconstruct the temperature structure along the transect. While these two models made use of simulated τ values the same methods could have been applied using τ as measured by PIES if sufficient inverted echo

sounders had been available along the full transect. For the August 1993 data the GEM-based simulation showed a good degree of agreement with the CTD-section with the exception of near the surface where the GEM-field failed to capture seasonal effects. Over much of the two sections the temperature difference is less than 0.25°C and only reaches 1°C near the thermocline while below the 300 dbar level the rms difference between the CTD-measured field and the GEM based recreation was 0.15°C. While Meinen and Watts (2000) did not provide similar figures for the November 1994 section they did note that the GEM-based simulation did manage to capture several important features such as a thickening of a lens of warm water near the centre of the Mann Eddy and steep isotherms offshore of the eddy. Additional comparisons were made between temperature measurements made from current meter moorings within one kilometre of a PIES. Data from the moored temperature sensors and the PIES/GEM based predictions were averaged over a period one week to reduce the effects of small scale variability and the results were compared. Below 1500m the rms difference was approximately 0.1°C, while in the vicinity of the thermocline (<300m) the difference was about 0.7°C.

To further quantify how well the GEM performed Meinen and Watts (2000) calculated the ratio of noise variance to total variance. For this comparison three measures of variance were defined for T. Total variance, measuring the spread of measured temperatures about the mean GEM field temperature for a given pressure:

$$\sigma^2_{tot,T}(p) = \frac{\sum_{a=1}^{A}\left[T_{meas.} - \overline{T_G(p)}\right]^2}{A-1}$$

Noise variance, measuring the spread of the difference between measured temperatures and predicted temperatures:

$$\sigma^2_{n,T}(p) = \frac{\sum_{a=1}^{A}\left[T_{meas} - T_G(p,\tau_{sim})\right]^2}{A-1}$$

Signal variance, measuring the spread of T for simulated return times about the mean GEM field temperature for a given pressure:

$$\sigma^2_{G,T}(p) = \frac{\sum_{a=1}^{A}\left[T_G(p,\tau_{sim}) - \overline{T_G(p)}\right]^2}{A-1}$$

The fraction of variance accounted for by the GEM field was defined as:

$$F_T(p) = \frac{\sigma^2_{G,T}(p)}{\sigma^2_{tot,T}(p)} = \frac{\sigma^2_{tot,T}(p) - \sigma^2_{n,T}(p)}{\sigma^2_{tot,T}(p)}$$

For all these expressions A denotes the total number of CTD casts, T_{meas} denotes temperatures measured and T_G denotes GEM temperature predictions. Bars over a value indicate averages. Meinen and Watts (2000) also developed similar expressions for δ.

Between 100 and 1500 dbar the GEM accounted for more than 95% of the variance. Between 1800 and 2500 dbar, this figure dropped to about 80% before steadily declining to below 20% near 4500 dbar. Calculations were also conducted for the specific volume anomaly (δ) yielding similar results.

A Circumpolar GEM:

In contrast to Meinen and Watts' GEM of the NAC , Sun and Watts (2001) made use of surface geopotential height instead of acoustic return time. This paper describes two GEMs: a 2D GEM along the WOCE SR3 line and a second low resolution 3D circumpolar GEM.

The 2D GEM of the WOCE SR3 line made use of six oceanographic transects along the WOCE SR3 line (45S to 56S) between 1991 and 1996. The 'geopotential height' (or more correctly, the dynamic 'height'), was taken between the surface and the 3000dbar level such that:

$$\phi_{3000} = \int_0^{3000} \delta dp$$

This value was calculated for each CTD cast based on the T and S profiles obtained from that cast. All property values at each pressure level were then plotted against ϕ_{3000} and a cubic smoothing spline was fitted to the data. From this the GEM fields $T_G(p, \phi)$, $S_G(p, \phi)$ and $\delta_G(p, \phi)$ were generated. From these fields it was possible to simulate a hydrographic section and compare it with real measurements (Figure 2.5). While the simulated section filtered out many small scale features it did succeed in recreating most frontal and mesoscale features.

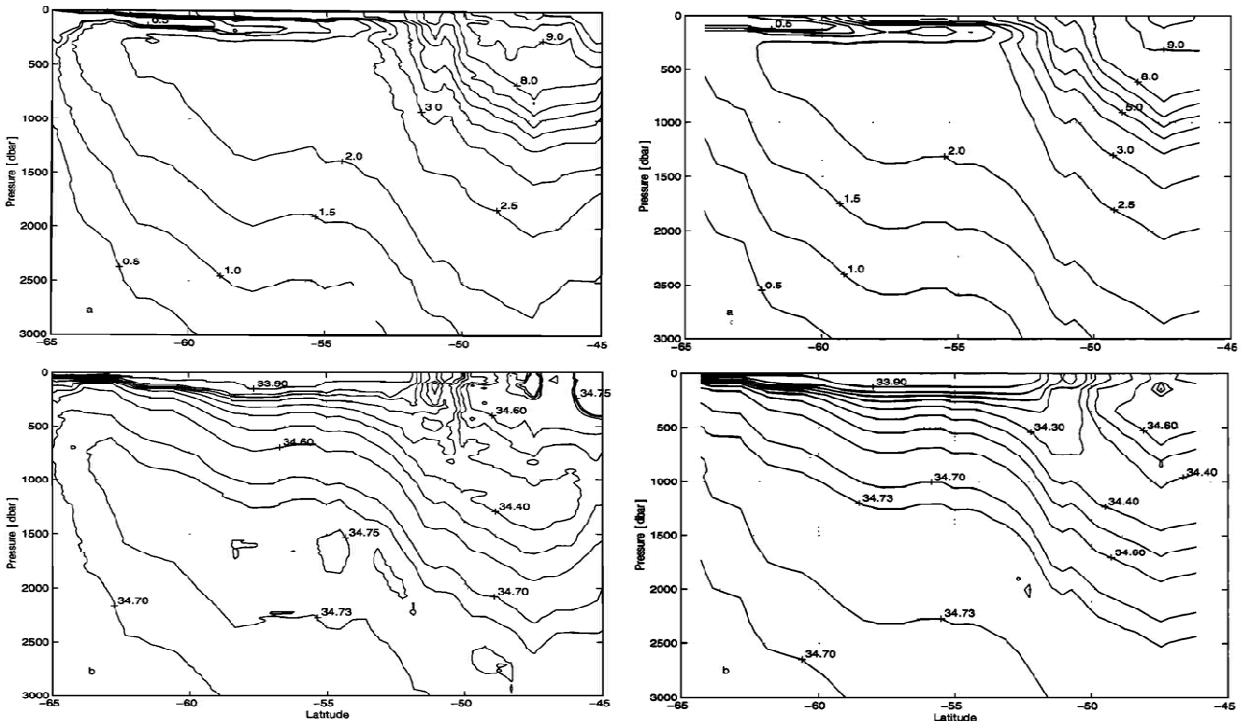


Figure 2.5: *Real (left) and simulated (right) temperature (°C, top) and salinity (psu, bottom) sections along the WOCE SR3 line. From Sun and Watts (2001).*

While the 2D SR3 line GEM succeeded in capturing most of the structure of the transect across the ACC, it is probable that if the same fields were applied on transects across another section of the ACC it would yield poor results. Thus Sun and Watts modified their GEM further by introducing longitude (denoted as λ) as a third dimension.

To adapt the GEM for longitude Sun and Watts made use of a historical dataset covering the ocean between 30°S and the Antarctic coastline between the years 1900 and 1990. Questions of the quality of early measurements subsequently reduced the usable time period to 1950 to 1990, and elimination of profiles which did not reach the 1000dbar level reduced the total number of stations to 9600. To produce the circumpolar GEM fields a set of 2D GEM models were produced using a similar methodology to the SR3 line. In this case the geopotential height between 100 and 1000dbars was used, rather than the 0-3000m interval employed in the SR3 GEM, in an attempt to maximise useable data and reduce seasonal variations. Each 2D GEM was separated by 5° of longitude and in order to include sufficient data points (200-300 casts) made use of CTD casts within a 'data window' 15° either side of the GEM (with the exception of near the Drake passage where higher data density allowed the use of a ±5° window). The rms residual of S and T fields between a depth of 300 and

1500 dbars (Figure 2.6) shows a general trend of larger errors for higher dynamic height (generally found in the north of the range).

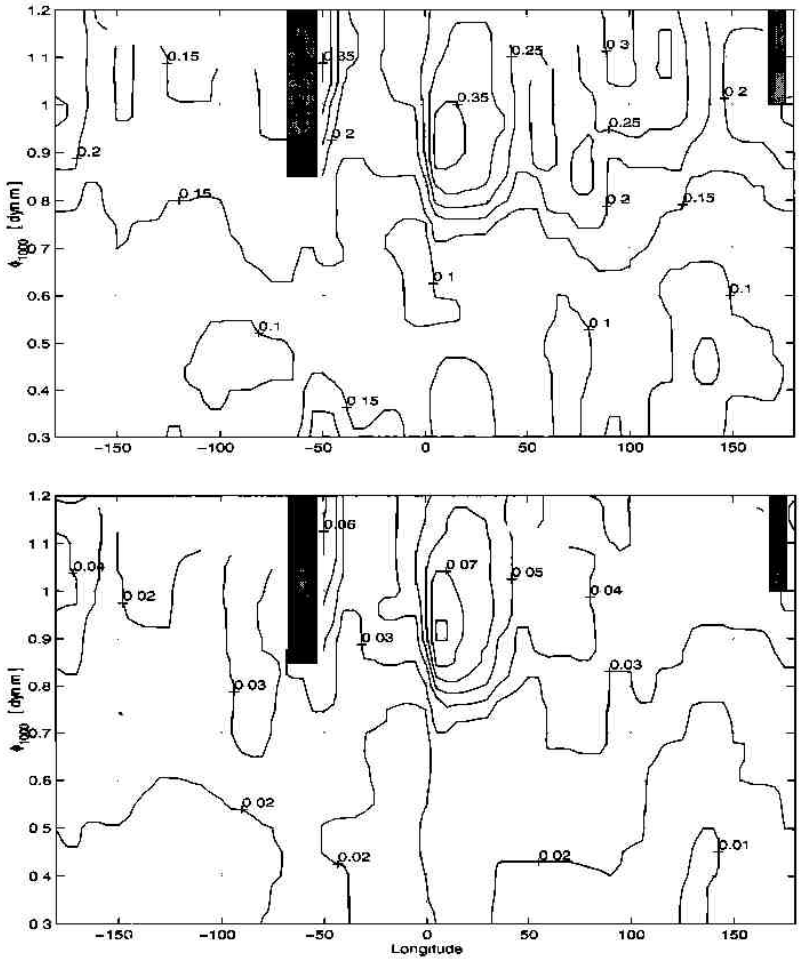


Figure 2.6: *Temperature residuals (°C, top) and salinity residuals (psu, bottom) in ϕ –longitude space. Higher ϕ values generally occur in the Northern regions of the Southern Ocean. From Sun and Watts (2001).*

Additionally, two regions of particularly large errors were identified: 50°W and 0°-30°E. In both regions the rms errors were almost three times the typical value of the circumpolar band. Sun and Watts (2001) speculated that these results arose from western boundary currents in the two regions introducing waters which were warmer and saltier than the ACC waters. This could have resulted in these bodies of water possessing different T and S profiles for similar ϕ-values, emphasising one of the major weaknesses of the GEM method. Finally, the proportion of variance in T, S and δ accounted for by the GEM fields was calculated in a similar manner to Meinen and Watts (2000). In the 300-3000dbar range over 97% of T and δ variance was described by the GEM fields; the proportion of salinity variance attributable to the GEM in the 200-1300dbar range was slightly lower at 95%.

A 2D GEM Of The Subantarctic Front:

At the same time as developing the circumpolar GEM discussed above Watts, Sun and Rintoul (2001) applied Meinen and Watts (2000) acoustic return time GEM technique (with the addition of a seasonal model to improve performance in the upper 300m) to a region of the Southern Ocean spanning the Subantarctic front (SAF). In addition they also implemented a GEM parameterization based on geopotential height. The creation of the geopotential height based GEM was not discussed in detail but was probably similar to the approach applied by Sun and Watts (2001) to the circumpolar GEM. Construction of the acoustic return time GEM made use of three data sources: 142 T&S profiles from eight CTD transects along the SR3 line; Time series data from two lines of Inverted Echo Sounders (IES) each spanning 270km across the Subantarctic Front and a set of 102 hydrocasts of the upper 300m, covering a region between 45°-55°S and 120°-160°E and spanning the period from 1956 to 1981. The 'main' SAF model used a similar method to Meinen and Watts (2000) NAC GEM with the addition of an empirical seasonal model which was used to remove seasonal variation from the data.

Watts, Sun and Rintoul (2001) found that the GEM temperature, salinity and density anomaly fields contained all the major structural features found across the SAF. Between 150 and 3000 dbar the GEM fields encapsulated 97% of temperature variance and 96% of specific volume anomaly variance. The GEM performed somewhat less well with regards to the variance in salinity, explaining only 93%.

2.5 A High Resolution 3D GEM of the Southern Ocean

The studies discussed above showed that dynamic height (or a suitable proxy such as geopotential height) and acoustic return time based GEM methods could be applied to certain regions of the Southern Ocean but it was only recently (Meijers et al. 2009b) that a high resolution three dimensional, ocean-wide GEM has been developed.

Data Preparation

In order to construct their Southern Ocean GEM Meijers at al.(2009b) started with 93000 ship-based CTD and hydrographic bottle stations drawn from the WOCE Atlas Database and 58877 Argo float profiles from the region between 35° and 66°S. The WOCE data were further sub-sampled to only include stations which included 20

measurements with at least one above 100dbar and one below 2000dbar. The Argo data were also sub-sampled to only include profiles with adequate resolution in T & S profiles, a quality control flag of 1 and a maximum depth of at least 1900dbar. This reduced the data to 16432 WOCE profiles and 14413 Argo profiles. Both WOCE and Argo profiles were then interpolated onto regular pressure levels.

In order to minimise errors caused by the convergence or proximity of water masses with similar dynamic height but different structures, these sets of profiles were further sub-sampled to remove any from north of the Subtropical Front (STF, at 200dbar T greater than 12°C and S greater than 35psu). This further reduced the number of available profiles to 24571. Finally, the remaining profiles were subsampled to include an equal number of observations from each month in order to reduce the summer bias seen in historical data. This left 15912 profiles for the creation of the GEM while the remaining 9659 were available to validate the GEM-based reconstructions.

GEM Field Creation

In contrast to previous methods (Sun and Watts 2001) which employed a set of meridionally oriented 2D-GEMs drawing T and S data from surrounding longitudinal data windows Meijers et al. (2009b) used objective mapping techniques to map T and S data onto regular grids in longitude and dynamic height space. This objective mapping was performed on 36 pressure levels from the surface to 5400dbar. For this mapping dynamic height was defined relative to the 2000dbar level as:

$$\phi_{2000} = \int_{2000}^{100} \delta dp$$

The upper limit of 100dbar was chosen to minimise the influence of the seasonal thermocline. This lower limit of 2000dbar gave a range of dynamic heights along the southern limit of the area of interest (~66S) of 0.5 to 1.9 dynamic meters. From this and the requirement to adequately resolve frontal features a dynamic height spacing of 0.01m was chosen. For the latitude-longitude grid a uniform spacing of 1/3° was selected.

The a priori error was calculated to provide a measurement of the noise inherent in the data. By assuming the oceanic noise was uncorrelated this error was estimated by Meijers, Bindoff and Rintoul (2009b) to be:

$$\sigma^2_{apriori} = \lim_{\varphi \to 0} \frac{1}{2} \left\langle \left(T_i - T_j \right)^2 \right\rangle$$

Where Ti and Tj indicate temperatures at two adjacent 'stations' and ϕ represents the 'distance' between the stations (in this case a dimensionless value between two points in dynamic height-longitude space).

Following a similar method to that adopted by Watts, Sun & Rintoul (2001) an empirical model was also fitted to seasonal variations in the upper 300dbar of the water column. Values obtained from this were then subtracted from the original data and the above process repeated to generate the final GEM fields. This substantially improved the near-surface performance of the GEM.

Performance Of The GEM Fields

Meijers, Bindoff and Rintoul (2009b) found that the GEM T and S fields accounted for most of the variance as well as capturing major features of the Southern Ocean. Several 'slices' across the fields were produced at specific longitudes (for an example see Figure 2.7), all of which showed a cooling and freshening of circumpolar deep water east of the Drake Passage and a strong AAIW fresh tongue.

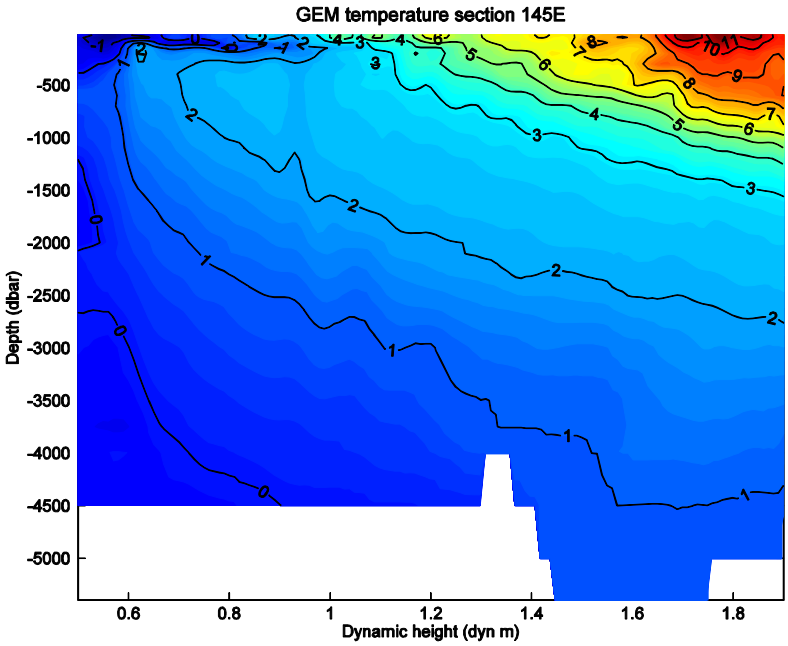


Figure 2.7: *Cross section of GEM T field at 145E. From Meijers et al. (2009b).*

While these fields do not contain information on the latitudinal location, the tendency for smaller dynamic heights to be found to the south and the relationship between specific values of dynamic height and oceanic fronts allowed Meijers et al. (2009b) to

identify a temperature inversion near the Polar Front and regions of deep mode water extending to depth north of the SAF.

The residuals between the hydrographic data not used during the creation of the GEM and collocated GEM predictions were plotted as a function of depth. Despite the occasional extreme outliers in the data generally the T rms residual in the upper 300dbars lay between 0.25 and 0.9°C; the residual reduced at greater depths, reaching 0.2°C below 500dbar and 0.1°C below 1000dbar. Above 300dbar the salinity errors ranged from 0.045psu to 0.103psu and at depths greater than 500dbar S errors were 0.025psu or less. These values generally lay close to the a priori errors.

The percentage variance in T and S captured by the GEM fields is shown in Figure 2.8.

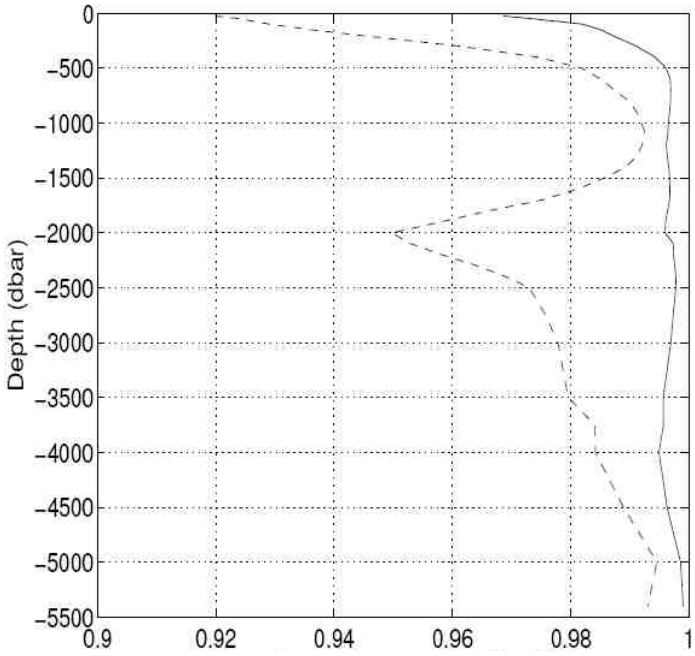


Figure 2.8: *Fraction of captured by the static GEM T (solid line) and S fields (dashed line) by depth. From Meijers et al. (2009b).*

Over 97% of total T variance is explained by the GEM near the surface and beneath 400dbar this figure approaches 99%. The GEM fields performed less well with salinity, near the surface only 92% of the variance is explained. This figure reaches a local maximum of about 98% between 500 and 1500dbar before dropping back to about 95% at 2500dbar, at greater depths the percentage of salinity variance explained increases again.

Combining the GEM Fields and Satellite Altimetry

In principle combining the GEM fields and sea surface data from satellite altimetry to create time evolving T and S fields should be relatively easy. In practice it is surprisingly difficult: Satellite altimetry gives the variation of SSH relative to a long-term mean rather than the dynamic or geopotential height. In principle it would be possible to convert from one to the other if the geoid height was known, however the geoid is not known to sufficient accuracy for this method to be useful. As a result Meijers et al. (2009b) had to generate an empirical relationship between SSH and ϕ by comparing in situ hydrological profiles to altimetry from the same location and time.

To do this Meijers et al. (2009b) made use of two additional data sources: The AVISO delayed time maps of mean sea level anomaly (MSLA) and the CSIRO Atlas of Regional Seas (CARS). AVISO combined altimetry data from four satellite missions to produce 728 weekly maps (with a 1/3° by 1/3° grid) of MSLA over the period from 1st October 1992 to 20th September 2006. CARS is based on a number of datasets including ARGO, World Ocean Database 2001 and WOCE WHP 3.0. Data derived from these sources were mapped onto a 1/2° by 1/2° regular grid.

CARS was used to produce a Mean Dynamic Topography (MDT) from dynamic heights. The MDT data were then interpolated to the locations of all available hydrographic data and the resulting values subtracted from measured in situ data to produce steric anomalies (pressure or density driven anomalies in sea surface height, which are associated with dynamic 'height'). These steric anomalies were then compared with AVISO SSH data observed at the same time and position. A linear equation was then fitted to the data:

$$y = 0.52x + 0.0017$$

Where y is the steric height anomaly and x is the observed SSH anomaly. Meijers, Bindoff and Rintoul (2009b) state that "The near 2:1 gradient is due to the limited range of integration (100-2000dbar) of the in situ anomalies, whilst the satellite observes the anomaly due to the full depth integration". This SSH anomaly to steric anomaly relationship does not change significantly across the longitude and latitude range covered by Meijers' GEM.

This relationship was then applied to the entire AVISO dataset before the results were added to the CARS MDT. This produced a dynamic height value at each altimetry grid point for all 728 'snap shots'. From these dynamic height values T and

S profiles were calculated from the GEM fields. As the GEM fields were created from T and S profiles which had been filtered to remove seasonal signals the seasonal trends in the near surface layers of the 'reconstructions' had to be added back in using an empirical model. An example of a hydrographic section and a SatGEM reconstruction is shown in Figure 2.9.

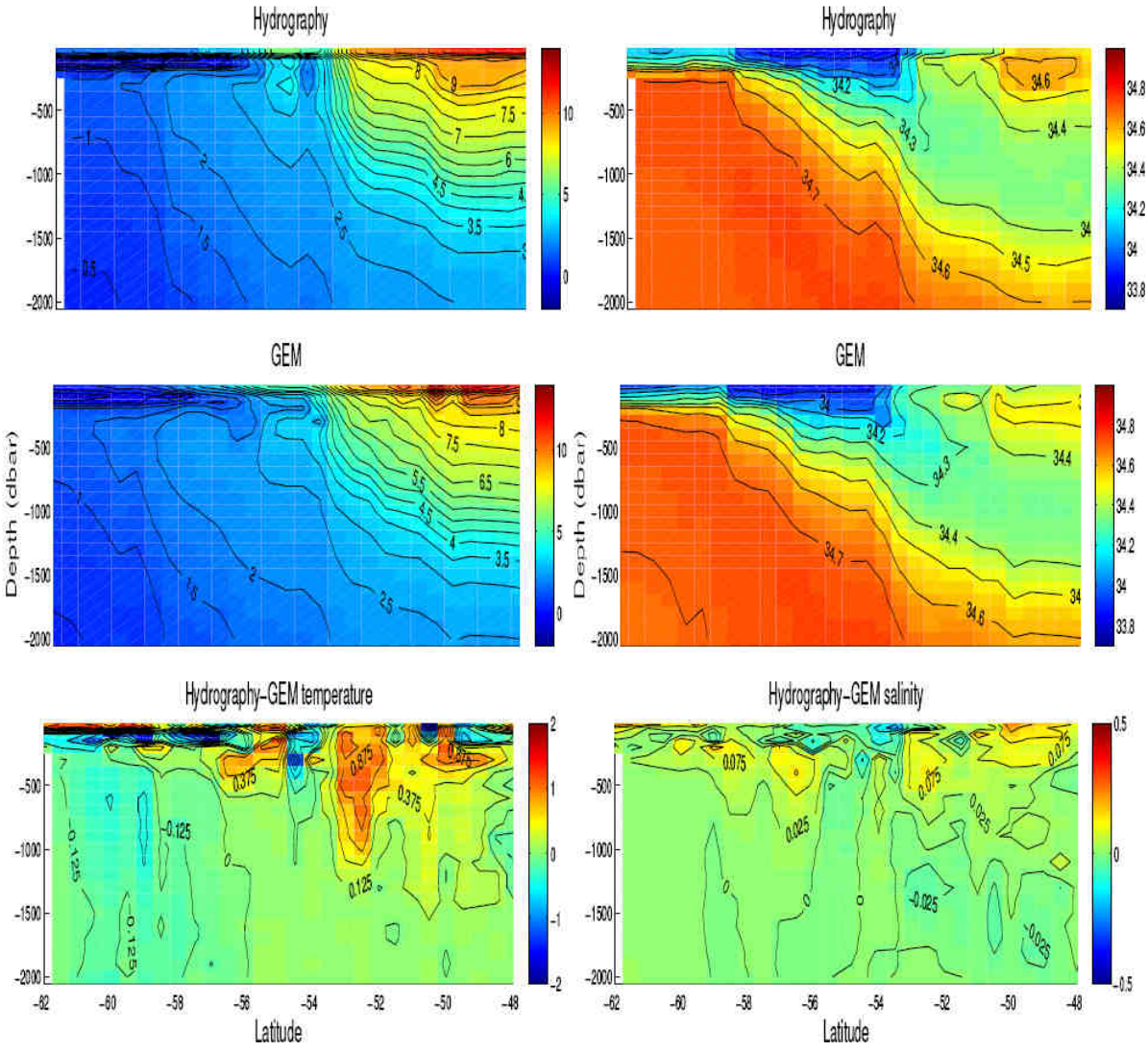


Figure 2.9: *Real and simulated temperature (left) and salinity (right) sections for the WOCE P16S survey line and the difference. From Meijers, et al. (2009b).*

The Time Varying T and S Fields

Meijers et al. (2009b) tested the accuracy and utility of the time varying T and S fields (henceforth, referred to as SatGEM fields) by examining the SatGEM's ability to reproduce independent hydrographic observations and recreate observed frontal positions in the waters south of Australia.

While the SatGEM T and S rms errors were typically larger than the residuals displayed by the static GEM T and S fields and the a priori error, the errors were still small. In the upper 300dbar the T rms error ranged between 0.6 and 1.16°C before decreasing to 0.45°C near 500dbar and 0.11°C beneath 1500dbar. The salinity error varied from 0.132psu at the surface to under 0.03psu beneath the 1000dbar level. Despite the slight decrease in performance relative to the static GEM fields and the a priori error, on average these estimates remained more accurate than estimates based on the CARS climatology (Figure 2.10).

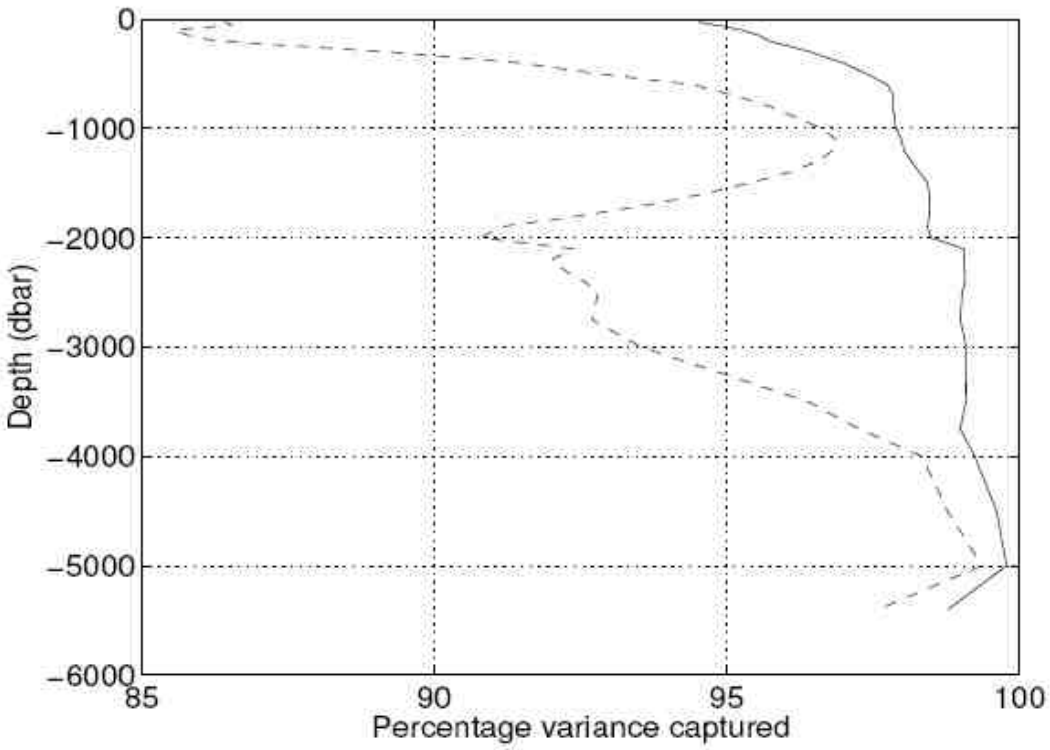


Figure 2.10: *Percentage variance captured by time varying (SatGEM) T and S fields. From Meijers, et al. (2009b).*

The percentage of T and S variance captured by the SatGEM fields exceeded 95% and 90% respectively over the entire water column except near the surface where the salinity figure drops to 86% (Figure 2.5.5). This represents a moderate decrease in performance relative to the static GEM fields.

The ability of the SatGEM to reconstruct hydrographic observations was tested against an independent T and S section taken during 2005 along the WOCE P16S line (150°W). The SatGEM section succeeded in recreating many of the major features (Meijers et al. 2009b) including: sharp temperature changes associated with the northern and southern SAFs; a temperature inversion associated with the Polar Front (PF) and a low salinity 'tongue' of Antarctic Intermediate Water (AAIW). The SatGEM section also proved to have some limitations. A warm anomaly at the 400dbar level near 55°S was only partially recreated in the simulated sections while the SatGEM produced weaker T and S gradients across the northern SAF than the hydrographic section. Similarly, the SatGEM section did not fully capture the shallowing of isotherms north of 49°S. Despite these issues Meijers et al. (2009b) concluded that the SatGEM did "... a very good job of recreating the section". When compared to reconstructions based on CARS and the World Ocean Atlas 2005 climatologies the SatGEM section identified the frontal regions with greater accuracy and also had smaller variations from the observed temperatures.

The time varying T and S fields were subsequently used by Meijers et al. (2009a) to calculate produce time varying fields of geostrophic velocities. The details of this are addressed in Section 3.

Use of SatGEM Fields to Identify Fronts

Meijers et al. (2009b) applied the SatGEM fields to the region of the Southern Ocean to try to identify key frontal features. Front were identified using the technique described by Sokolov and Rintoul (2007), by defining a front as a region with a SSH gradient above a particular value. This value was taken as 0.25 m/100 km in Sokolov and Rintoul (2007) and 0.3 m/100 km in Meijers et al. (2009b). Sokolov and Rintoul (2007) also observed that these zones of high SSH gradient generally are associated with particular contours of SSH.

After identifying a front by SSH gradient and determining the associated SSH contours for the region between 130° and 160°E, temperature and salinity data from the SatGEM fields at each SSH value was compared to hydrographic frontal definitions. Meijers et al. (2009b) found that with the exception of the Northern branch of the SAF frontal features defined by SSH coincided well with the hydrographic definitions. In the case of the Northern SAF while the SatGEM fields

typically produced higher temperatures (by 0.5-1°C) than corresponding definitions of the front there remained a significant overlap.

Chapter 3: Datasets

3.1 The Velocity Dataset

Developing the Velocity Fields

From the time varying T and S fields Meijers, at al. (2009a) were able to create density and geopotential height anomaly fields. Assuming purely geostrophic driven transport the coriolis force balances forces due to pressure gradients (Stewart 1997) and thus the velocity of a parcel of water at any given moment will be tangential to pressure contours and so, also to density and SSH contours. Thus, the relative latitudinal and longitudinal velocity components (v and u) could be calculated using the 'thermal wind' equations:

$$v = -\frac{1}{f}\frac{\partial \phi_{surf}}{\partial x} \qquad u = \frac{1}{f}\frac{\partial \phi_{surf}}{\partial y}$$

Where f denotes the coriolis parameter, ∂x and ∂y denote the latitudinal and longitudinal separation between grid points and ϕ_{surf} denotes the dynamic height as referenced to the sea surface as defined by the Mean Dynamic Topography. As these velocities are based on dynamic heights relative to the surface, in order to obtain absolute velocities Meijers et al. (2009a) also calculate the surface geostrophic velocities and added the resulting values to the relative geostrophic velocity fields for all pressure levels.

To account for wind driven effects the Ekman velocity was calculated by taking the weekly average u and v components of the wind speed at 10m altitude. The wind velocities were interpolated onto the same grid as the geopotential velocities and then converted to equivalent frictional velocities and then into wind stresses. From these wind stresses the Ekman transport across the faces of each cell in the velocity grid was calculated. By assuming the Ekman transport was confined to the upper 75 dbar of the ocean and that 64% of the transport occurred in the surface to 25 dbar cell and 36% in the 50dbar cell, Meijers et al. (2009a) were able to convert the Ekman transport values into Ekman velocities. These values were intended for use in calculating bulk transport properties, and reflect the average Ekman velocity. As a result the Ekman spiral is not resolved and so the 25 dbar and 50 dbar velocities are rotated by approximately 90° from the actual Ekman velocity at the surface.

Performance of SatGEM Velocity Fields

As a test of the velocity dataset Meijers et al. (2009a) compared estimates of the average velocity of an Argo float with a parking depth of 1000dbar over one cycle with SatGEM velocities at the mid-point of the direct path between the Argo float's descent and surfacing points. From this Meijers et al. (2009a) determined correlation coefficients on a region by region basis. The longitudinal correlation coefficients ranged between 0.49 and 0.72 with a global value of 0.58. The latitudinal values were lower, ranging from 0.30 to 0.62 with a global value of 0.48.

The Dataset

The velocity dataset produced by Meijers et al. (2009a) was provided as 38 matlab files, two corresponding to Ekman velocities at 25 and 50dbar and 36 corresponding to geostrophic velocities at pressure levels between 25dbar (approximately 25 meters) and 5400dbar (~5400m). Between 200dbar and 2500dbar these depth levels are spaced at 100dbar intervals; above 200dbar the spacing is 50dbars with the exception of the 25dbar level. At depths greater than 2500dbar the separation becomes more irregular. Each of these matlab files contains two 728 by 187 by1079 matrices holding the u and v velocity components. The 'axes', or co-ordinate system, of each matrix correspond to the date, latitude and longitude respectively. A section across one of these matrices at a given time value produces a velocity field for that date and depth.

For the geostrophic velocity files the velocity values are given in cm/s with west and south taken as positive. The Ekman velocity files give values in m/s with north and east positive instead. Both the Ekman and geostrophic velocity models assign values of 'not a number' (NaN) to any points at which velocities could not be calculated (e.g. regions with a 'depth' of less than 2000dbar). All velocities are mapped onto a 1/3° by 1/3° grid. The latitude axis runs from 69.94°S to 34.88°S and the longitude axis spans from 179.50° W to 179.83°E. The time axis used for both the Ekman and geostrophic velocity fields has the same seven day divisions covering a period from 1st October 1992 to 20th September 2006 as the S and T fields from which the geostrophic velocity is calculated.

3.2 Argo Data

Following the receipt of data from an Argo float, the data undergoes a number of tests to filter bad data such as incorrect dates, locations, T, S and velocities (defined as greater than 3 m/s) as well as unusually large spikes in values before being released to realtime datasets. Further tests, corrections and manual inspections are applied before the release of the delayed time data. Details of these procedures are discussed in Wong et al.(2009). The data is generally released in netCDF by one of two global data centres (GDACs).

Generally, there are at least four files associated with an Argo float:

- <float id>_meta - general float information.
- <float id>_prof - data from all profiles to pass quality control.
- <float_id>_tech - technical information for each cycle and trajectory.
- <float_id>_traj – trajectory data such as surface fixes and times.

In some cases files containing individual profiles are also stored. In addition to the float specific files the GDACs also hold a three 'directory files' in ascii format, detailing general data on single profile files and float trajectory and meta files. Of these files only the Meta files, trajectory files and trajectory directory file are of direct relevance to this experiment. Full details of the file formats can be found in the Argo Data Management: User Manual (Carval et al. 2008).

The Argo data used with the Meijers et al. (2009a) velocity model was obtained from a digital library operated by the Tasmanian Partnership For Advanced Computing (TPAC) at the University of Tasmania.

GDAC Directory Files

The directory file used as part of this project was sourced from the French GDAC Coriolis (http://www.coriolis.eu.org/cdc/default.htm). The trajectory directory file contains data on all floats accessible through the DAC. This data is formatted as a comma separated list, with each row laid out as follows:
file, latitude_max, latitude_min, longitude_max, longitude_min, profiler_type, institution, date_update

Variable names are generally self explanatory. 'File' specifies the location and name of the netCDF files. The latitude and longitude entries indicate the limits of the recorded float locations. Profiler_type contains a three digit code specifying the model

of Argo float or other sampling device used (see section 3.8 of Carval et al. (2008) for details). Institution contains a two letter code used to denote the institution that deployed the float in question. Date_update contains a 14 digit number specifying the date on which the trajectory netCDF file was updated (the right most four digits specify the year, the next two digits specify month, then day and so on).

Meta File

General data on a given float is contained within the Meta file. Full details of the contents and format of the Meta file can be found in section 2.4 of Carval et al. (2008). An Argo meta file consists of six sections: definitions; general information; float characteristics; deployment and mission information; sensor information; calibration information and trajectory information. The definitions and general information sections are also found in the traj file.

The definitions section consists of values defining the number of different cycle types programmed into the float; the number of parameters recorded and technical information like the number of bytes allocated to storage of strings or dates. The general information section contains such details as when the meta file was created; when it was updated; etcetera. The float information specifies the model of Argo float and details of the data transmission and positioning systems. The deployment and mission section contains information on the time and location the float was deployed and the float's current status. The sensor and calibration sections describe the sensors fitted to the float; accuracy of those sensors and any pre-launch calibration performed on the sensors. Of most relevance to the model developed in this study is the data in the trajectory section. This section contains variables specifying the periods the float is programmed to spend in a given phase of a cycle; the parking pressure and the maximum pressures encountered while descending and while profiling. Data from each of the meta files used in the model are specified in Table 3.1

Variable	Description	Units
CYCLE_TIME	Total time for one entire cycle	Hours
PARKING_TIME	Time spent at parking pressure	Hours
DESCENDING_PROFILING_TIME	Time spent descending to parking pressure	Hours
ASCENDING_PROFILING_TIME	Time spent profiling back to surface from DEEPEST_PRESSURE	Hours
SURFACE_TIME	Time spent on the surface	Hours
PARKING_PRESSURE	Pressure at which float 'parks' at between the end of a descent and the start of a profile	dBar
DEEPEST_PRESSURE	Deepest pressure sampled	dBar
LAUNCH_LATITUDE LAUNCH_LONGITUDE	Location at which the float was launched	degrees

***Table 3.1**: Data from Argo meta files used within the model*

Trajectory File

Like the meta files, traj files can be sub-divided into several sections: definitions; general information; locations/measurements; cycle information and history information. The definition and general information sections are essentially the same as in the meta files. The location/measurements section contains dates, locations and cycle number of recorded transmissions. The cycle information records data on the dates and times of the start and end of each phase of each cycle. The history section records previous changes to the file. Dates and times in the locations and cycle information sections are expressed as decimal days since a reference date, usually the 1st of January 1950. Data from each of the traj files used are listed in Table 3.2

Variable	Description	Units
LATITUDE, LONGITUDE	Arrays containing latitudes and longitudes of surface fixes	Degrees Degrees
CYCLE_NUMBER	Array specifying cycles numbers for surface fixes	
JULD	Julian dates of surface fixes	Day
JULD_ASCENT_START, JULD_ASCENT_END	Julian dates for start and end of ascent for each cycle	Day
JULD_DESCENT_START JULD_DESCENT_END	Julian dates for start and end of descent for each cycle	Day
JULD_START_TRANSMISSION	Julian dates for the start of data transmission for each cycle	Day

***Table 3.2:** Data from Argo traj files used within the model*

Chapter 4: Model Development

4.1 Model Development and Implementation

Over the course of its development the numerical model developed as part of this study went through a number of iterations. The initial intention was to make use of Matlab's default streamline function to trace a float's trajectory. The presence of multiple pressure levels and the time varying nature of the velocity fields ultimately rendered this option impractical. As a result, it was decided to instead calculate the trajectory by assuming that for a sufficiently short time step the displacement of a float can be considered as linear. Code for this purpose was developed in Matlab.

As many of the names of routines and variables utilised in the Matlab code are similar, typographic conventions have been adopted for the rest of this thesis to discriminate between them. Matlab functions or routines are named in ***bold italics***; variables or data loaded from Argo netCDF files are named in CAPITALs; working variables are shown in *italics* and output variables in *underlined italics*.

The numerical model was initially implemented by writing two functions, the first served as a harness routine (***single_float***) which managed the velocity fields and called the second function as needed; the second function (***parking_depth***) calculated the displacement for each time step. This initial implementation was limited to a single pressure level and used step-wise transitions between velocity fields. Subsequently additional functions were added to handle transitions between pressure levels (***descend*** and ***profile***) and related modifications were made to the harness function. Next linear interpolation between the weekly velocity fields was introduced. This entailed considerable changes to the routines ***parking_depth***, ***descend***, ***profile*** and ***single_float***. In order to reduce the duplication of code within ***single_float*** the process of updating date and time variables was passed off to a new function, ***time_elapsed***.

With the step-wise velocity field transitions it was practical to use an entire velocity field for the displacement calculations. However, after implementing the temporal interpolation this approach resulted in an excessively long runtime for each time step. The solution to this was to introduce sub-sampling of the velocity field. While this improved run-time considerably it increased the complexity of ***single_float***

and forced the introduction of another function, ***gen_lon_axis***, to handle the creation and management of longitude axes needed for the interpolation.

The model was then tested with a wide range of Argo data and a number of problems were identified. The majority of these issues were associated with unexpected features in the Argo data such as: invalid, null or illogical entries in variables such as CYCLE_TIME and JULD_ASCENT_START; or variables not actually stored in a given netCDF file. It was possible to develop methods to work around the majority of these issues. For example, in the event that individual descent, drift and profile times could not be calculated from the JULD variables in the traj file for a given float cycle the code would default to using the programmed times listed in the meta file. If that also failed (as happened with a number of floats) the model would then apply a generic division of the cycle times.

The final stage of development involved the adaptation of the model to automatically load and run a large batch of floats. This involved developing another function (***model_main***) to control the loading of SatGEM velocity matrices and to feed the files paths the location of trajectory and meta files of each float within TPAC's library into ***single_float***.

4.2 Model Components

The numerical model consists of five major and a couple of minor routines. The major routines are discussed in detail below and the relevant code is provided in Appendix A. Code and brief descriptions of the minor modelling functions can also be found in Appendix A. Routines were also developed to ease the data processing and visualisation but these are not included.

Model Main

Model_main serves as the initialisation script for the entire model. It does not take any input arguments. Upon start-up it loads two .mat files: paths.mat and GEM_msla_axis.mat. The former contains a set of cell arrays each of which lists file paths for floats which underwent a particular 'type' of cycle (e.g. 1000dbar drift followed by 2000dbar profile). The latter contains the latitude, longitude and time coordinate system (or axes) used during the creation of the SatGEM velocity datasets.

Model_main then proceeds to generate the velocity field axes (*center_lon*, *center_lat*) from the axes contained within GEM_msla_axis.mat before also

generating a 429 element vector (*gemdate*) containing the number of days elapsed since the start of the SatGEM fields for each time 'slice' of the SatGEM velocity matrices. Next the function loads the velocity matrices of the surface (approximated with the 25dbar level as no 0dbar level was avalible), parking pressure and profiling pressure for the current batch of floats. Then the function enters the first of a series of while loops: for each pass through the loop ***model_main*** feeds the general data (velocity fields, coordinate systems, etc.) and one file path into ***single_float***. Once all floats paths in a given cell array are done the while loop terminates; velocity matrices are loaded for new pressure levels and the function enters the next while loop, which repeats the process outlined above.

Single Float

Single_float serves as the harness function for the trajectories of an individual float. The input arguments accepted are listed in Table 4.1. An overview of the algorithm applied in ***single_float*** is shown in Figure 4.1.

Arguments	Format	Description	Units
center_lon	1079 element vector	Longitude axis	degrees
center_lat	187 element vector	Latitude axis	degrees
gemdate	429 element vector	Time axis	Julian days since start of GEM
cycles	scalar	Maximum number of cycles to run (used in testing and debugging)	
u_surf, u_park, u_deep	429x187x1079 matrix	Velocity matrices	cm/s
v_surf, v_park, v_deep	429x187x1079 matrix	Velocity matrices	cm/s
file_path	string	File path specifying location of Argo data	

Table 4.1: *Input arguments required by* ***single_float.***

Upon launch the function loads the meta and trajectory files for the input float. From this it extracts the parking pressure; deepest pressure; surface locations (and associated cycle numbers and times); programmed and estimated or recorded times spent in each phase of a cycle. Next, where possible ***single_float*** generates vectors containing the time spent by the float in each phase for each cycle. Then the velocity

field just before the float's launch is identified and the offset in hours between the float launch and the velocity field is calculated. Both these values are used to create the two element vector *date_time_index* (weeks since start of GEM; hours since most recent velocity field was loaded). After creating NaN filled arrays to store locations, status and cycle number at all time steps the function enters the main while loop which operates until either the cycle index (*i*) exceeds the input argument *cycles* or when the start date of the current cycle comes within three weeks of the end of the GEM coverage.

Upon entering the while loop ***single_float*** finds values within CYCLE_NUMBER which equal the cycle index. If no such values are found the cycle index is incremented until such a value is found and then *date_time_index* is recalculated. Upon identifying such a value the function proceeds to load the start co-ordinates into the vector *curloc* which specifies the current location of the float. Next, the model checks if the location lies within the latitude range covered by the SatGEM fields (i.e. between 35S and 70S), in the event it does not the function increments the cycle index and returns to the start of the while loop.

If the location does lie with the valid latitude range ***single_float*** proceeds to identify the nearest values in *center_lon* and *center_lat* to the start location and then creates auxiliary latitude and longitude axes. Typically, these axes span 5 degrees either side of the location both in latitude or longitude. Special cases apply if the location is within 5 degrees of the boundaries of the GEM fields: the latitude axis is reduced so as not to run beyond the latitude range, while two longitude axes are created both covering the range 160E to 160W.

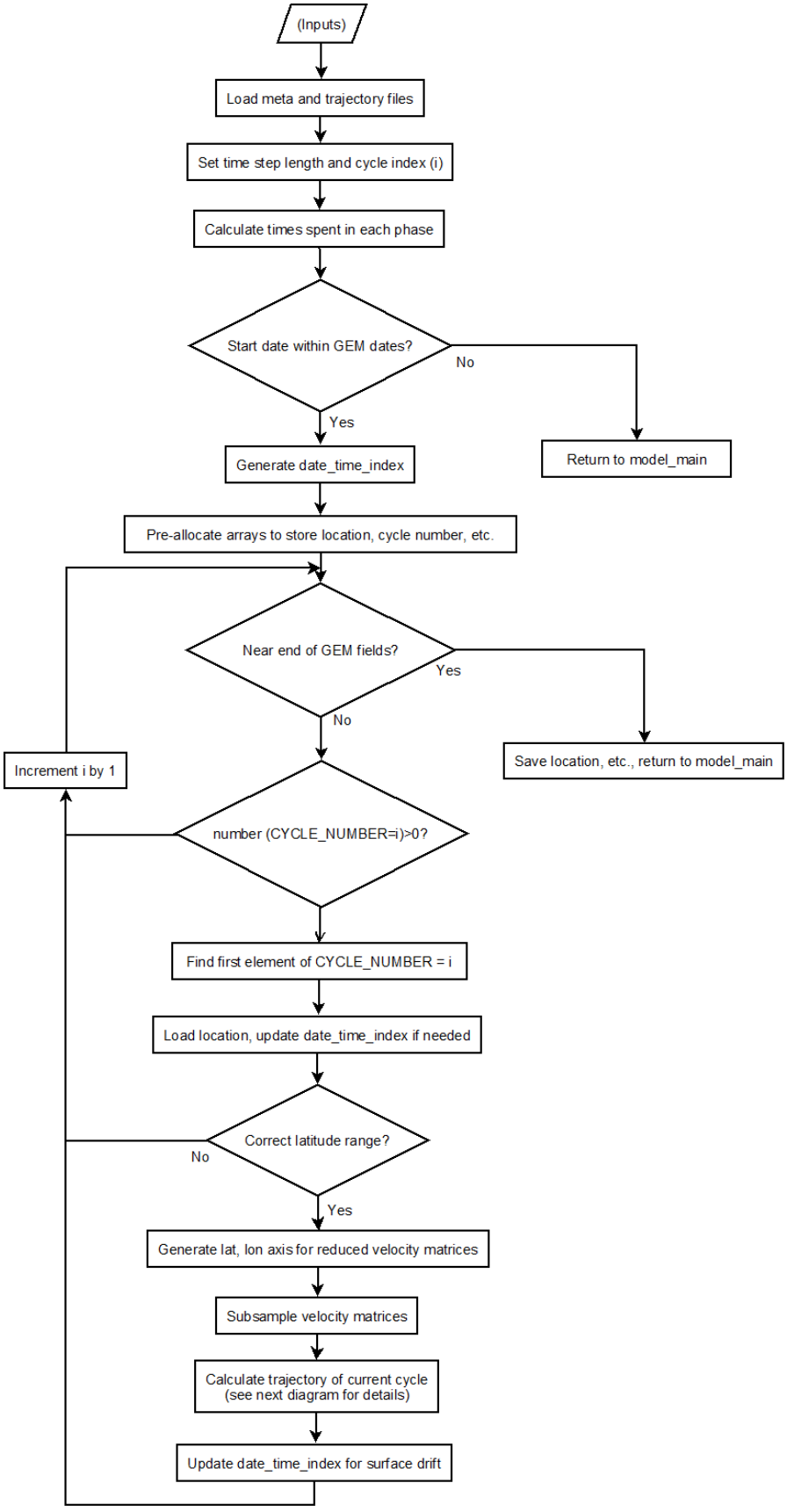


Figure 4.1: *Flow chart showing the algorithm used in single_cycle*

The first of these considers the entire longitude range in degrees East (i.e. treats 160W as 200E) while the second does so in degrees west (i.e. treats 160E as 200W). These longitude axes are used to enable a float to 'wrap' back around from one end of the longitude range to the other. For example, if the float starts at 179E when the float passes 180E the longitude axis is switched. This procedure is necessary to work around the limitations of Matlab's interpolation functions. Next the velocity matrices are subsampled to produce smaller matrices consisting of three velocity fields with limits defined by the auxiliary axes.

At this point the ***single_float*** commences calculating trajectories. This process calls three similar sections of code and functions for the descent, drift at parking depth and profiling phases. The algorithm common to all these sections is outlined in Figure 4.2.

This process starts by identifying the time limit on the time to be spent in this particular phase. This value is obtained from one of the following: the entry in the associated time vector calculated earlier; the time specified in the meta file or a fraction of the total cycle time derived from a generic cycle outlined in Carval et al. (2008). Then the function enters a while loop which runs while the time elapsed during the current phase is less than or equal to the time limit. Within this while loop the function selects the appropriate auxiliary longitude axis and then calculates the next location. Then the time elapsed within this phase is incremented by the length of a time step and the location is stored within the NaN arrays mentioned above. *Date_time_index* is then updated and the process either returns to the start of the while loop or moves onto the next stage. Upon exiting the main while loop the trajectory data (location, corresponding cycle number and 'status') are saved to a file (<argo float number>.mat) and ***single_cycle*** terminates.

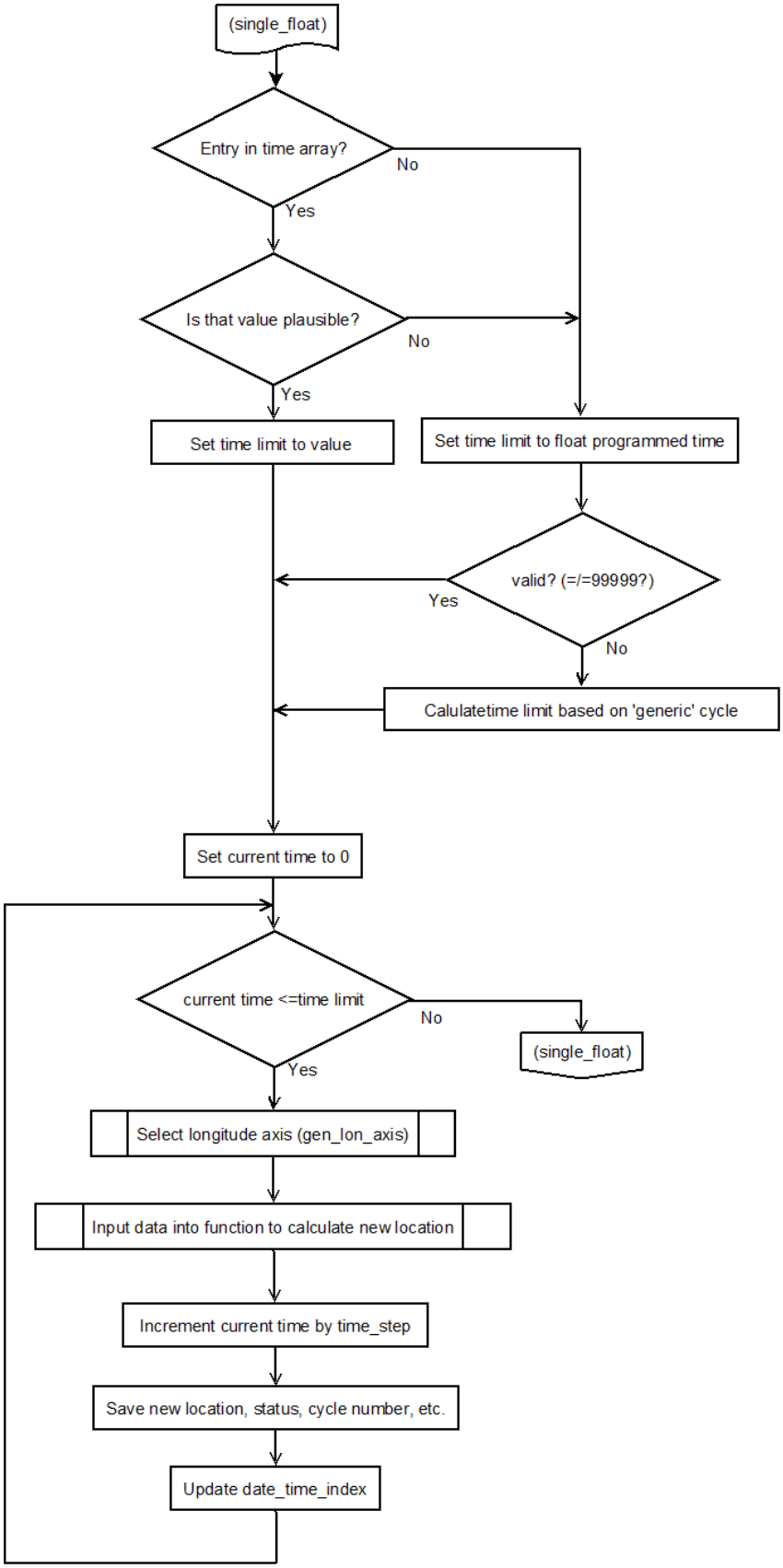


Figure 4.2: *Flow chart showing details of the single_float algorithm as used to calculate trajectories for each phase.*

Descend

The function ***descend*** returns the location of the float at the end of a time step while the float is sinking from the surface to the parking pressure or from the parking pressure to the deepest pressure of the profile. A general overview of the algorithm common to the functions ***descend***, ***parking_depth*** and ***profile*** is shown in Figure 4.3. Input arguments for the function are specified in Table 4.2.

Arguments	Format	Description	Units
curloc	Three element vector	First Element: Latitude Second Element: Longitude Third Element: Status (NaN if previous velocity was invalid)	degrees degrees
timestep	Scalar	Length of time step	hours
date_time_index	Two element vector	First element: number of weeks since start of GEM fields. Second element: hours since most recent velocity field loaded.	weeks hours
lat	Vector	Latitude axis for velocity fields	degrees
lon	Vector	Longitude axis for velocity fields	degrees
time_axis	Vector	Time axis for velocity fields in weeks since the start of GEM.	weeks
u4d1, u4d2	3d matrices	Reduced longitudinal velocity fields.	cm/s
v4d1, v4d2	3d matrices	Reduced latitudinal velocity fields.	cm/s
time	Scalar	Time since start of descent	hours
time_sink	Scalar	Time taken for full descent	hours
z	Scalar	Change in ‘depth’ over descent	dBar

Table 4.2: *Input arguments for the function* ***descend****.*

Descend calculates the current depth of the float and the current time in weeks since the start of the GEM. This conversion of the time is necessary as the variable time_axis which gives the time values of the velocity fields (each of which are ‘slices’ of the SatGEM velocity matrices) used in the interpolation, are in a ‘weeks since GEM start’ format. The input location is checked to ensure it lies within the latitude range of the GEM fields. If so, the process continues as outlined below, if not the interpolation is skipped the velocity components are set to 0.

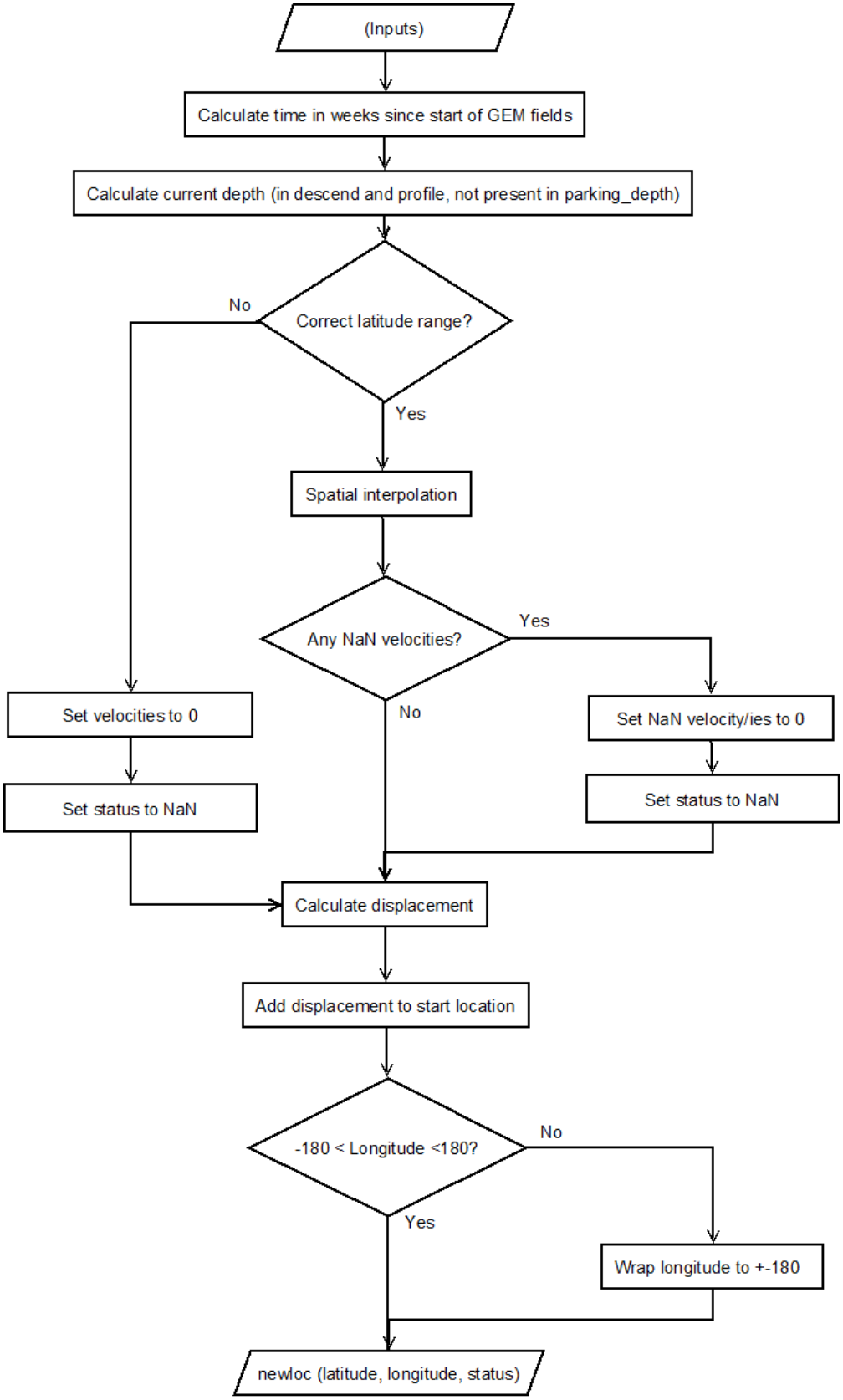


Figure 4.3: *Flow chart showing the algorithm used in descend, parking_depth and profile.*

Descend then interpolates the v and u components of the upper and lower velocity fields for the current time and location using Matlab's Interp3 function. If any of the four resulting velocity components returns a NaN, that value is set to 0 cm/s for further calculations and status is set to NaN. Next the velocity components at the current depth are calculated using linear interpolation (specifically Matlab's Interp1 function) and the results are used to calculate the displacement of the float. This displacement is converted into degrees of latitude and longitude using a function sourced from the CSIRO's Sea Water toolset. These displacements are then added to the location to *curloc*(1) and *curloc*(2). The output is a variable *new_loc* which has the same format as *curloc* (i.e. three element vector: latitude, longitude and status). If the longitude component of *new_loc* (*new_loc*(2)) gives a value of, for example, 185W then the longitude is converted to the corresponding longitude in 180W to 180E (in this example to 175E) before the result is returned.

In principal Descend could easily be modified to use more than two depth levels. This was not done for two reasons. Firstly, memory constraints meant it would not be practical to have more than three or four depth levels pre-loaded and any attempt to load up additional depth levels as needed would have had a major negative effect on runtime. Secondly, as an Argo float would typically only spend about 10 hours out of a 10 day cycle descending, the errors in displacement due to this approximation are unlikely to be significant when compared to the displacement over the period spent at the parking pressure.

Parking_Depth

Parking_depth is used to calculate the displacement of the float while at the programmed parking depth. Input arguments are listed in Table 4.3.

Arguments	Format	Description	Units
curloc	Three element vector	First Element: Latitude Second Element: Longitude Third Element: Status (NaN if previous velocity was invalid)	degrees degrees
timestep	Scalar	Length of time step	hours
date_time_index	Two element vector	First element: number of weeks since start of GEM fields. Second element: hours since start of current cycle.	weeks hours
lat	Vector	Latitude axis for velocity fields	degrees
lon	Vector	Longitude axis for velocity fields	degrees
time_axis	Vector	Time axis for velocity fields in weeks since the start of GEM.	weeks
u4d1, u4d2	3d matrices	Reduced longitudinal velocity field.	cm/s
v4d1, v4d2	3d matrices	Reduced latitudinal velocity field.	cm/s

Table 4.3: *Input arguments for the function* ***parking_depth****.*

As per ***descend***, ***parking_depth*** first calculates the time in weeks since the start of the GEM fields then checks the location lies within the valid latitude range. If so *u4d* and *v4d* are interpolated to determine the u and v velocity components at the location specified in *curloc*. If not, the velocity components are set to zero and status is set to NaN. If either velocity interpolation returns a NaN the corresponding velocity component is set to 0 and status is set to NaN. Finally, the displacement is calculated, converted to degrees and added to curloc to yield the new latitude and longitude. These values and the status are returned in the output argument new_loc. If the longitude component of *new_loc* (*new_loc*(2)) gives a value of, for example, 185W then the longitude is converted to the corresponding longitude in 180W to 180E (in this example to 175E) before the result is returned.

Profile

The function ***pofile*** calculates the new location for each time step during the ascent from the maximum depth to the surface. The input arguments are the same as ***descend***

with the exception of *time_sink* being replaced by *time_rise*. The algorithm is essentially the same as that used in ***descend***.

Time_Elapsed

The function ***time_elapsed*** is used to update the variable used to track the current time. This was done to avoid needlessly duplicating code at multiple locations in ***single_float***. The function requires two input arguments, *timestep* and *date_time_index*, and returns *date_time_index* as its output.

The function takes *timestep* and adds it to the second element of *date_time_index* (time in hours since the current velocity field was loaded). If this results in *date_time_index*(2) exceeding 168 hours then the first element of *date_time_index* (weeks since the start of the GEM fields) is incremented by one and *date_time_index*(2) is reset to 0.

4.3 Model Output

For each float the model saves the details of the simulation at each time step in a matlab file. This data is stored in a number of variables:

- Save_lat: Latitude of the simulated float at the end of each time step.
- Save_lon: Longitude of the simulated float at the end of each time step.
- Save_cycle: Cycle number
- Save_phase: Contains a number for each time step indicating which phase of a cycle the model was in. 1 corresponds to the start point, 2 to sinking to parking depth, 3 to movements at parking depth, 4 to descent to profiling depth and 5 to the profile itself.
- Save_status: 'Status' of the float. Contains a NaN for cycles in which a float wanders into areas in which the GEM fields contain no valid velocities.

In addition to the data produced by the model these files also contain the dates (in julian days since Matlab's default reference date) for the start and end of each cycle. These dates were extracted from float trajectory files as an additional step in processing after the model had been run for all floats.

In addition to the output files described above the model also produced a debug file for each float containing the following variables:

- debug_cycle_time: The times (in hours) for each time step relative to the start of the corresponding cycle.
- debug_data_time_index: The values of date_time_index for each time step.
- debug_dates: Indexes of the time-slices of the velocity fields sub-sampled for a given cycle number.
- debug_time_index: Record of values used within descend, parking_depth and profile for time-based interpolation. In effect the same as of date_time_index, except expressed purely in ‘weeks since start of SatGEM’ format. Used to ensure descend, parking_depth and profile were reading and processing date_time_index correctly.
- debug_velocities: Velocities (both u and v components) as calculated for each time step.

The contents of the debug files were only used during development; validation of the model and as a means of double checking issues with a small number of data points.

Chapter 5: Data Preparation and Model Validation

5.1 Data Pre-Processing and Selection of Model Settings

Argo Data Processing

Before running the model the Argo data had to undergo a degree of pre-processing. First floats for which maximum or minimum latitude values lay outside the limits of the velocity fields were filtered out, this left a total of 1110 Argo floats. These floats were then sub-sampled to remove both floats not present in TPAC's dataset and floats deployed after the end of the time period covered by the SatGEM fields. This reduced the number of usable floats to 326. Finally data in the Meta files were used to sort the floats by parking pressure and maximum pressure. This allowed the floats to be grouped according to the 'type' of cycle and thus made it possible to run the model for a batch of floats without reloading unnecessary depth levels from the velocity dataset for each run. Subsequent filtering for floats with multiple profile 'types' (e.g. two cycles drifting at 1000 dbar and profiling to 2000dbar followed by a cycle drifting at 1000dbar and profiling from 1000dbar) within the model itself further cut the number of usable floats to 255. The breakdown of these floats by cycle 'type' was:

- 183 float were programmed with a parking depth of 1000dbar and a profiling depth of 2000dbar.
- 28 with a parking depth of 1500dbar and a profiling depth of 2000dbar.
- 20 with both profiling and parking depths set to 1000dbar.
- 16 with a parking depth of 1900dbar and profiling depth of 2000dbar.
- 5 had a parking depth of 400dbar and profiling depth of 2000dbar.
- 4 with a programmed parking depth of 500dbar and profiling depth of 2000dbar.

As such, a total of 7 different pressure levels and the additional Ekman velocities at 25dbar were required to run the model, but as noted above memory restrictions meant only three pressure levels could be handled at one time.

Selection of Time Step Duration

The vast majority of parameters required by the model were set directly or indirectly by Argo data and the only variable which was required to be specified was the duration of one time step. A number of different time step periods of up to 2 hours

were considered. To determine a suitable value several tests were run (see Figure 5.1) in a vertically uniform and time varying field.

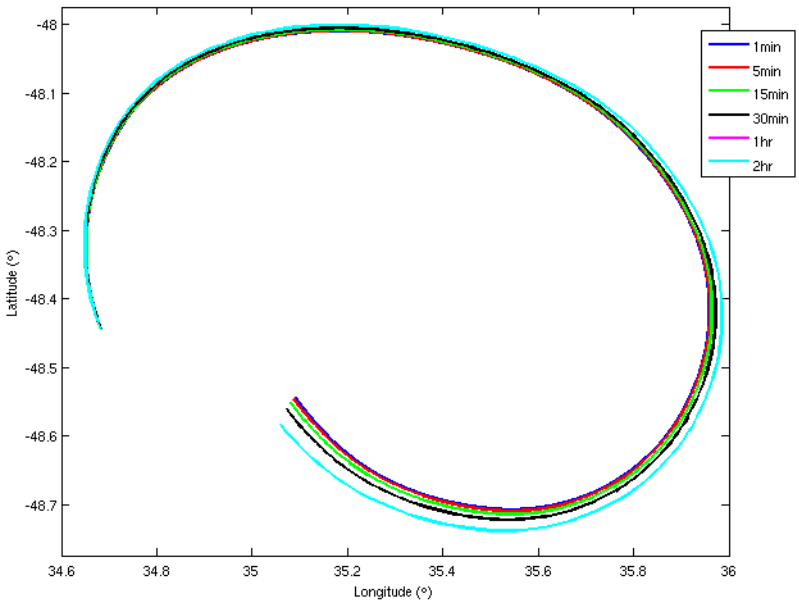


Figure 5.1: *Test trajectories for 1 minute, 5 minute, 15 minute, 30 minute, 1 hour and 2 hour time steps.*

In a region with a moderately complex velocity field for time steps of 30 minutes or less the trajectories for the different time steps remain in close proximity to each other, while for longer time steps the resulting trajectories diverge from the paths calculated with shorter time steps. Additionally, even in regions with simple velocity structure in which longer time step trajectories closely follow other the trajectories there is a tendency for time steps in excess of 1 hour to produce a significant overshoot. As such, 30 minutes represents the maximum time step from which reliable results can be produced, however it remains preferable to use a shorter time step. Runtime considerations served to eliminate the extremely short time steps (such as 1 minute or 5 minute periods). Based on the combination of those two factors the time step was set to 15 minutes.

5.2 Testing the Model

Before running actual Argo floats through the model to it was necessary to validate and test the model's performance. As such three major features of the model had to be tested: spatial interpolation between pressure levels; temporal interpolation

between ‘slices’ of the SatGEM fields and the ability of the model to trace a plausible path through a non-uniform velocity field.

Depth Interpolation

To test the interpolation between pressure levels two simple synthetic velocity fields were constructed and applied (1 cm/s south at the surface and profiling depth and 0 cm/s at parking depth) and the model was run. Each of the synthetic velocity fields was uniform and temporally static. As such, velocity values at a given depth could easily be calculated and then be compared with velocities stored in debug files by the model. The model was run for a total of 100 cycles, each with a fixed length of 168 hours (138 hours at parking depth and 10 hours in each of the descent, profiling and surfacing phases). A parking pressure of 1000dbar and a profiling pressure of 2000dbar were used. Figure 5.2 shows sample a plot of depth versus velocity for the descent phases of two profiles superimposed over a graph of the expected velocity:

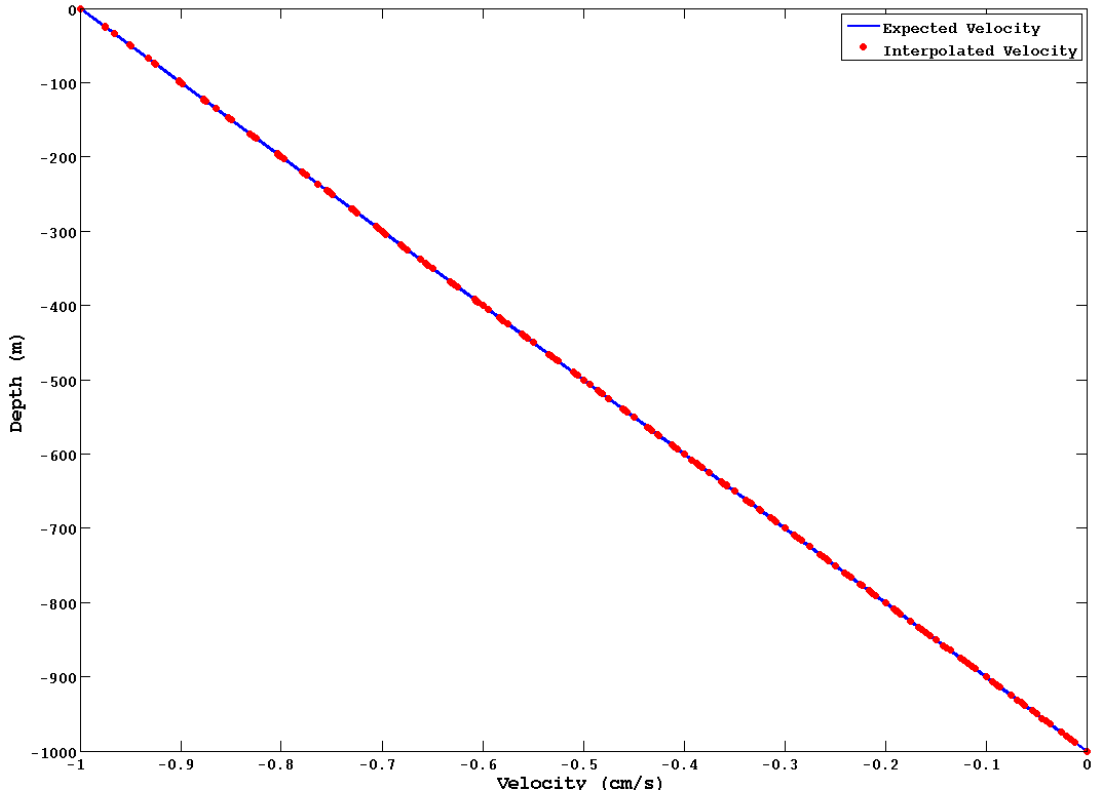


Figure 5.2: *Interpolated velocities vs depth for two descent phases (red dots) and expected velocities vs depth (blue line).*

This demonstrates that for these synthetic fields the model produces a good match with the expected results. This is supported by the residuals between expected and interpolated velocities (Figure 5.3) for all test cycles:

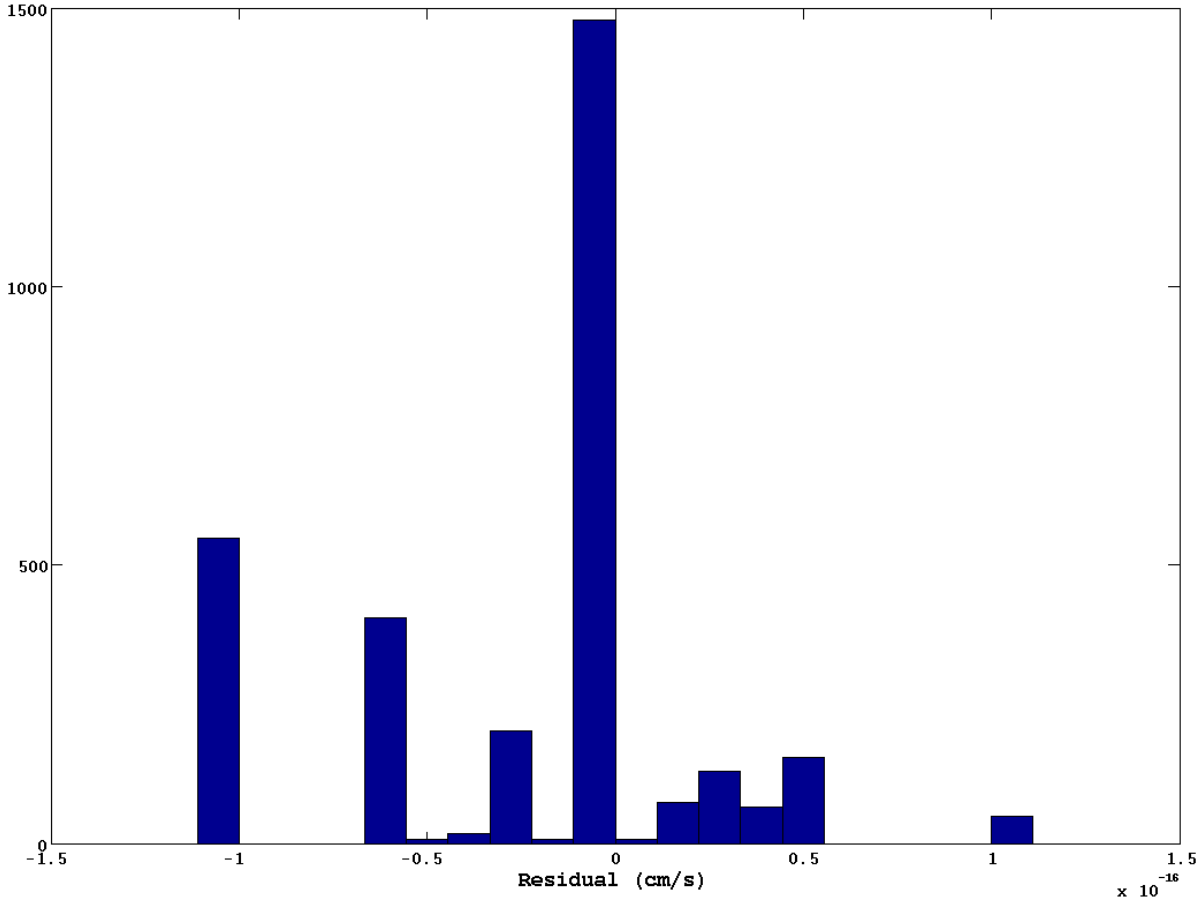


Figure 5.3: *Histogram of residuals between expected velocity and velocities produced by depth interpolation.*

All residuals are of the order of 10^{-16}cm/s. In other words, any errors introduced by the depth interpolation are likely to be insignificant

Temporal Interpolation

To test the model's temporal interpolation a synthetic field with spatially uniform but time-varying velocities (alternating linearly between 1 cm/s South and 0 cm/s at seven day intervals) was constructed and applied to all operating levels. The velocity for a given time between each weekly 'slice' of the velocity fields can be calculated with ease and then compared with data stored in debug files. The cycle length and operating depths remained the same as outlined above. As seen from the sample output in Figure 5.4 the interpolation procedure in the model produces highly accurate velocities.

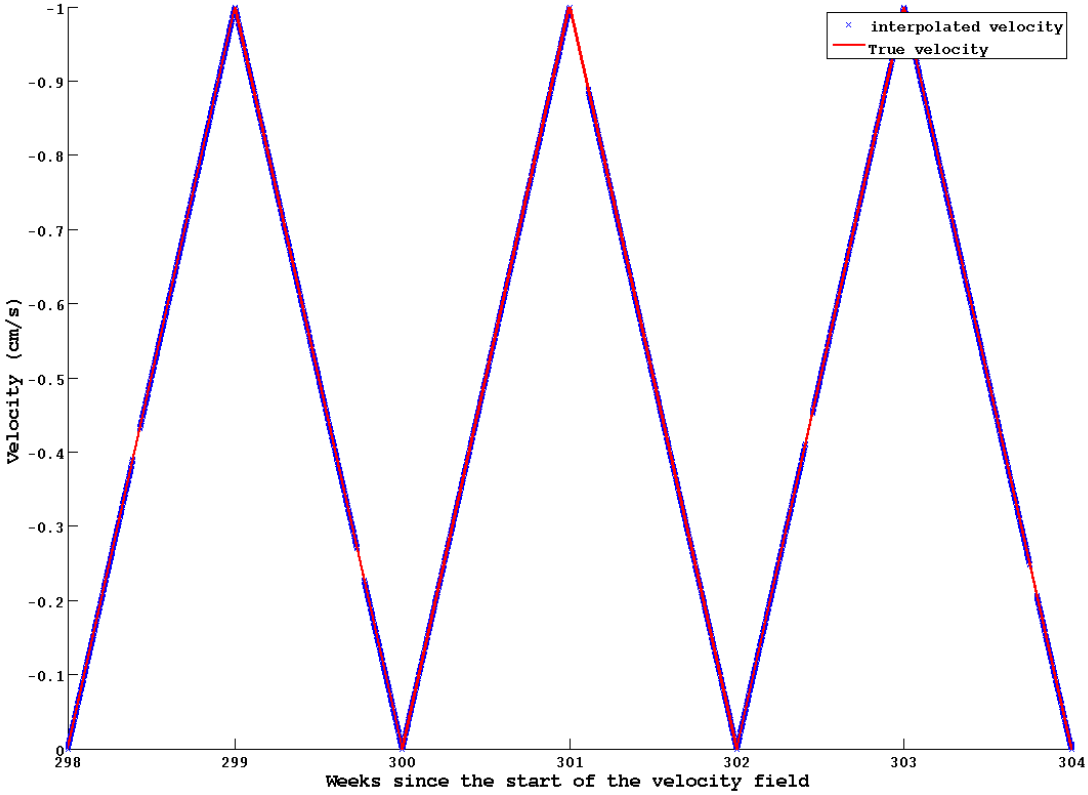


Figure 5.4: *Sample output from a test run of the model in a spatially uniform time-varying field. Gaps correspond to periods of surface drift which was not modelled.*

Displacement

In order to test that the model was producing correct displacements it was tested for a number of starting positions in three temporally static and spatially uniform velocity fields (1 cm/s south, 1 cm/s west and 1.41cm/s south-west). As the timing and velocities of each field are known, displacement could be independently calculated and compared with that obtained directly from the model by the use of the function ***sw_dist*** from the CSIRO's Seawater Matlab toolkit (http://www.cmar.csiro.au/datacentre/ext_docs/seawater.htm). Histograms of residuals between the modelled and true displacements are shown in Figure 5.5 for each of the three fields.

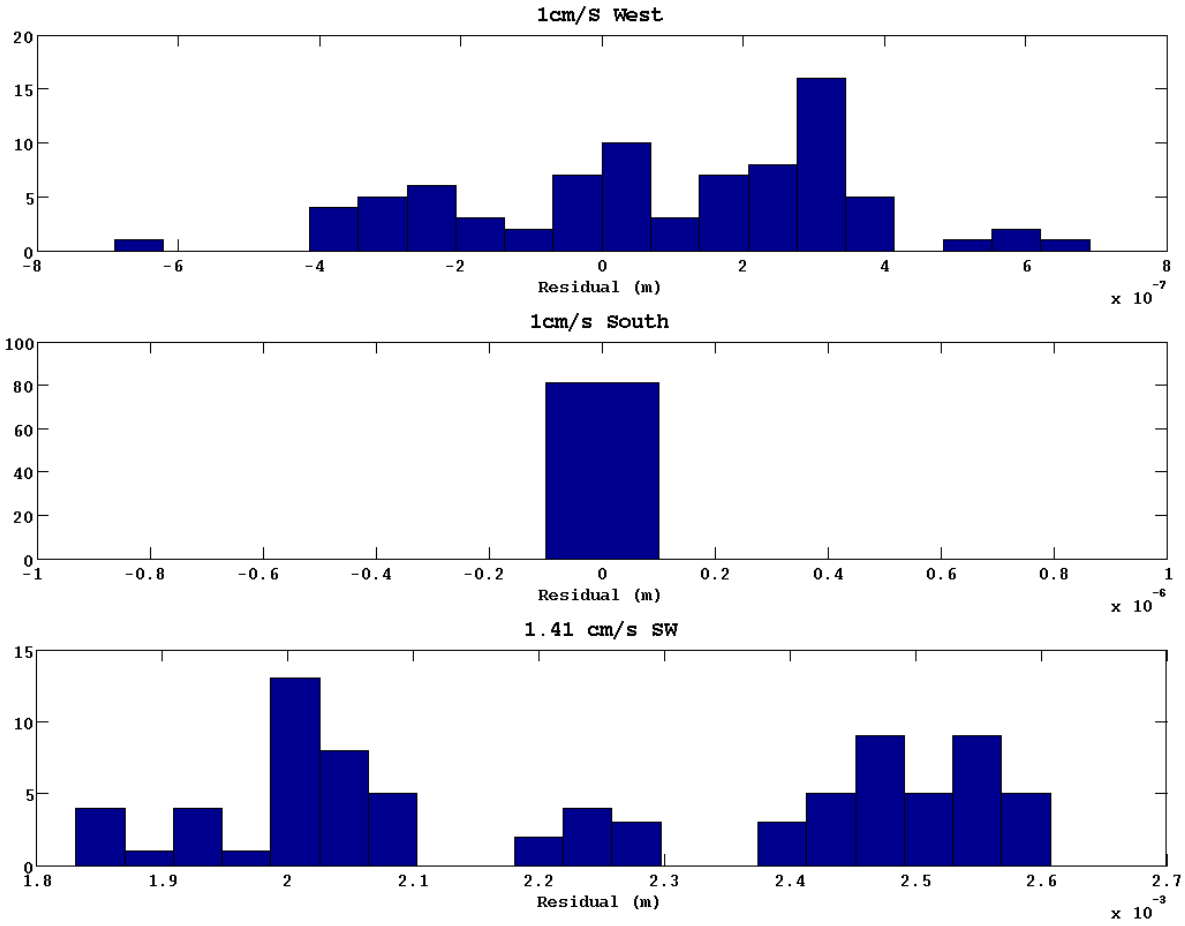


Figure 5.5: *Histograms of residuals between expected velocities and interpolated velocities.*

The errors in all cases are many orders of magnitude less than the path length. As such, it is clear that the numerical model produces accurate displacements.

Path Following

In principle the ability of the model to produce reliable latitudinal and longitudinal displacements should imply that the model has the ability to accurately track a path in a spatially varying velocity field. A second test of this was performed by running the model in a vertically uniform and temporally static field. The resulting paths were then visually compared to both the surrounding velocity field and Matlab's ***streamline*** function. Sample outputs are shown in Figure 5.6.

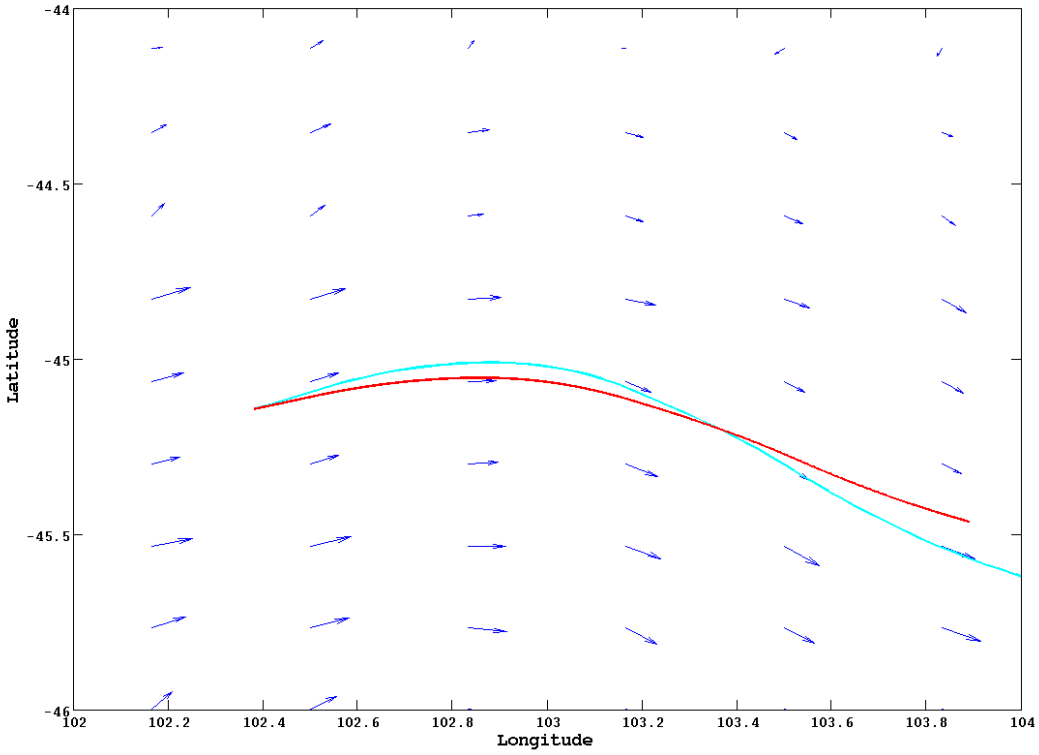


Figure 5.6: *One trajectory of the model (red) in a temporally static, vertically uniform but spatially varying velocity field compared with* ***streamline*** *output (light blue) for the same field.*

As can be seen in the sample output shown above, while the results do not always match up with the streamline path (a result in the differences in algorithms, ***streamline*** determines the direction to move then moves a fixed distance while the functions used within the model determine both a direction and a distance to move) the model does trace plausible paths through the velocity fields.

Chapter 6: Analysis Methods

6.1 Output Data Processing and Definitions

Before conducting the analysis the data produced by the model had to undergo processing. The majority of this processing was done within the function ***calc_stats***. This function calculates the misfits (ε) and path lengths (P_{Argo}, P_{Sim}) for each float in degrees of latitude and longitude before also converting the values into kilometres. As indicated in Figure 6.1 the path lengths P_{Argo} and P_{Sim} are defined as the distance between the descent location and the real and simulated surfacing locations respectively. ε is defined as the value of P_{Argo}- P_{Sim}.

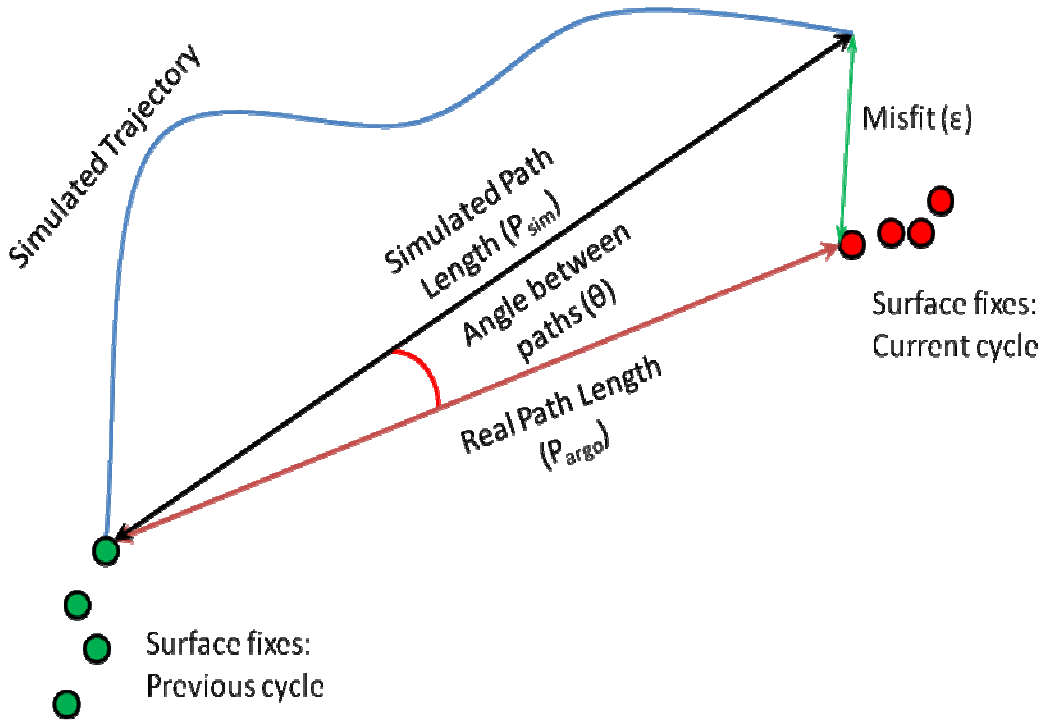


Figure 6.1: *Schematic diagram of Argo trajectory variables.*

From the misfit and path length displacements for a cycle a 'relative misfit' (the ratio of ε to P_{Argo}) was calculated. ***Calc_stats*** also extracted the temperatures at 400dbar and sea surface height anomalies for the descent, simulated surfacing and known surfacing positions from the applicable datasets. In addition the function ***find_depth*** was used to calculate the bathymetry at each location. Next the correlation coefficient and R^2 values were calculated for the entire population, individual floats and specific regions.

6.2 Correlation Coefficients and R^2 Values

The correlation coefficient and R^2 values serve to quantify the quality of a linear fit between two variables and the percentage of variance in one variable accounted for by the linear fit with the other variable.

For vector data such as most of the model output the correlation coefficient is defined as:

$$R = \frac{a \bullet b}{\|a\| \cdot \|b\|}$$

This can also be expressed as:

$$\frac{b^T \cdot a}{\sqrt{a^T a} \cdot \sqrt{b^T b}}$$

Where a and b are the variables on which the analysis is being conducted and ^{T} denotes the transposition of a vector.

The R^2 value is defined as the square of the correlation coefficient. Correlation Coefficients and R^2 values were calculated for the entire population; in a circumpolar array of 5° by 5° cells (on a cell by cell basis) and for specific regions. These values were calculated both using the equations specified above and Matlab's inbuilt function. In the event of there being slight differences between the values returned by the Matlab function and the equations listed above then the mean of the three values was used.

It is worthwhile noting that the function constructed to calculate correlation coefficients and R^2 for 5° by 5° cells filtered out any cells containing less than five data points in order to avoid the creation of high R^2 values in data deficient regions.

6.3 Relative Misfit and Relative Path Length

While misfits and path lengths provide a means to measure the performance of a float cycle there is a need for a consistent means of comparing misfits and path lengths for all floats. To do this two quantities were defined: relative misfit (ε_r) and relative path length (P_r):

$$\varepsilon_r = \frac{\varepsilon}{P_{Argo}} \qquad P_r = \frac{P_{sim}}{P_{Argo}}$$

In both cases the relevant values have been normalised by dividing by the real path length (P_{Argo}). Both relative misfit and relative path length were plotted against P_{argo}. Float cycles were classed into three categories based on each cycle's ε_r value and mapped. P_{argo}, P_{sim}, ε and ε_r were also mapped onto a ¼° by ¼° grid.

Note that unless stated otherwise for the rest of this study P_{argo}, P_{sim}, ε and ε_r denote the absolute value of the variable rather than components or vectors.

6.4 Variance and Histograms of ε, P_{sim} and P_{Argo} and θ

By comparing the variance and plotting the histograms of the latitudinal and longitudinal components of ε, P_{sim} and P_{Argo} in the same graph some judgements could be made as to the overall performance of the model.

By projecting ε, P_{sim} and P_{Argo} onto unit vectors parallel and perpendicular to the local SSH contour the variance of misfit and path length parallel and perpendicular to the contours can be determined. As variance of a sample is conserved the changes in these values relative to the latitudinal and longitudinal variances can be used to test if the performance of the model is better or worse along streamlines and as a preliminary means of investigating mixing or flow across SSH contours. This was implemented by calculating the mean SSH field for the three week period surrounding each float cycle, loading the SSH at the descent location of the cycle and then identifying the nearest grid point with SSH within ±0.005m of the descent location's SSH. A vector could then be produced between the descent point and the grid point and then be inputted into ***Project_misfit*** to determine components of misfit and path length parallel and perpendicular to the mean SSH contour. Due to potential inaccuracies arising from the means of identifying SSH contours this method was only applied to trajectories with path lengths of over 50km.

6.5 SatGEM Performance, Frontal Features and SSH

Two approaches were used to examine any relationships between frontal position and SatGEM. The first method was to create a circumpolar map of floats with particularly large and small relative misfits superimposed on a map of mean frontal positions derived from the mean SSH field. The SSH values used to define frontal positions were obtained by Meijers et al. (2009b) as described in section 2.5 (see Table 6.1). While the relationship between these SSH values and frontal features have been shown to apply with remarkable consistently across the region south of Australia

(Sokolov and Rintoul 2007) may not hold for the entire circumpolar range. None the less, it is a more convenient means of defining frontal positions for this study than the hydrography based definitions.

Front	Northern SAF	Middle SAF	Southern SAF	Northern PF	Southern PF	Southern sACCF
SSH(m)	1.60	1.36	1.21	1.02	0.81	0.7

Table 6.1: *Peak SSH values associated with frontal features. From Meijers et al. (2009b).*

The second approach involved sorting the data into a number of bins and then for each bin calculating the ratio of the number of float cycles with small ε_r to total cycles in the bin and the number of floats with large ε_r to total float cycles. Both these values were plotted against SSH.

6.6 Depth Relationships

Relationships between model performance and the depth of the ocean were investigated by first plotting relative misfit against the depth and then by sorting the misfits into a number of bins based on depth and for each bin calculating the ratio of the number of float cycles with small ε_r to the total number of floats in the bin and the number of floats with large ε_r to the total number of floats. Both these values were plotted against depth.

6.7 Temporal Relationships

Analysis of variation in model performance as a function of time was conducted in a similar manner to the analysis of model performance as a function of depth (Section 6.6) with time series of misfit and relative proportions of high and low misfit cycles plotted for data binned according to time.

6.8 Comparison with Random Paths

In order to assess the ability of the SatGEM fields to predict the path of an Argo float better than by chance alone synthetic random dataset was constructed. This dataset was created for each descent by location producing a random displacement at a random angle and from these generating a random simulated ‘surfacing’ location. The direction was generated from a uniform random distribution across 360° while the

distance was generated from a Gaussian distribution with negative values removed and a mean and standard deviation matching that of $|P_{Argo}|$. Analysis of misfit distribution, standard deviation, correlation coefficients and R^2 values for the random dataset were conducted using the same methodology as the model data. Additionally, after mapping to a grid of 5°x5° cells the ratio of the difference between simulated and random path R^2 values was calculated for each cell and normalised by the random path R^2.

Chapter 7: Results

7.1 Example Trajectories

The primary output of the numerical model is the simulated trajectory of an Argo float through the 4D SatGEM velocity dataset. These paths can be visualised and qualitatively assessed by plotting the Argo float descent and surfacing locations together with the simulated trajectory of the float onto a vectorial representation of the velocity field at the parking depth of the float. Due to the temporally varying nature of the SatGEM velocity fields these data were plotted with respect to the average velocity fields for a three week period spanning the cycle of the Argo float. The velocity averaging process accounts for the minor discrepancies in the apparent trajectory of the float relative to the velocity vectors since the trajectory is actually calculated from velocities calculated by temporal interpolation of the SatGEM dataset.

To demonstrate the primary model output three trajectories are illustrated in Figures 7.1, 7.2 and 7.3. These three simulated trajectories were selected for display as representative of paths with low, medium and large relative misfits respectively.

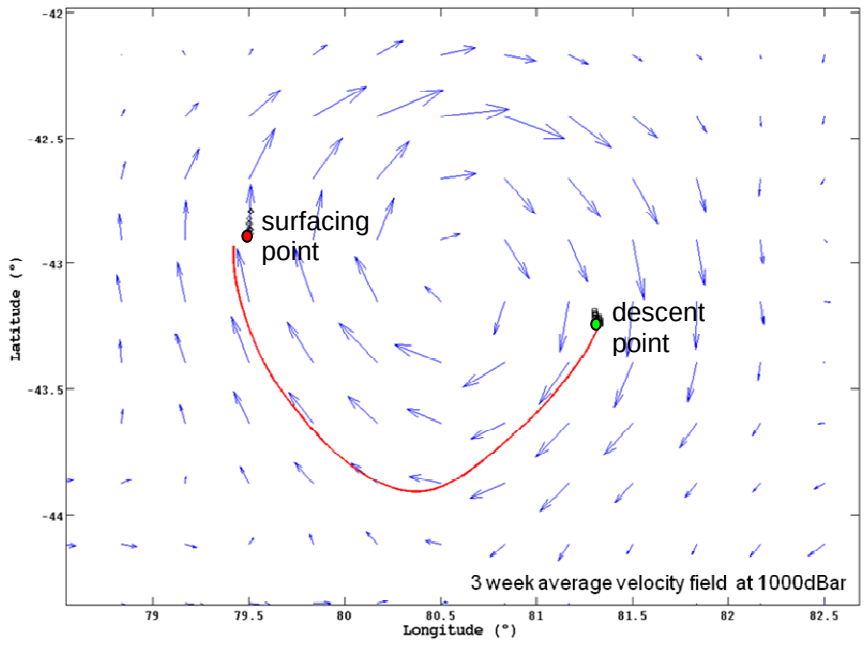


***Figure 7.1**: Example of a simulated Argo float trajectory in a relatively simple velocity field with corresponding small misfit in surfacing position.*

Figure 7.1 shows a simulated trajectory in a relatively simple velocity field for which the simulated and actual surfacing locations are close together and hence the absolute and relative misfits are small. Figure 7.2 shows a trajectory with moderate misfit in a relatively simple velocity field for which the direction of the simulated path is in reasonable agreement with the actual surfacing location but the path length is less than the actual distance traversed by the float indicating that the SatGEM velocity field in this case successfully predicts the direction of motion but underestimates the actual velocity.

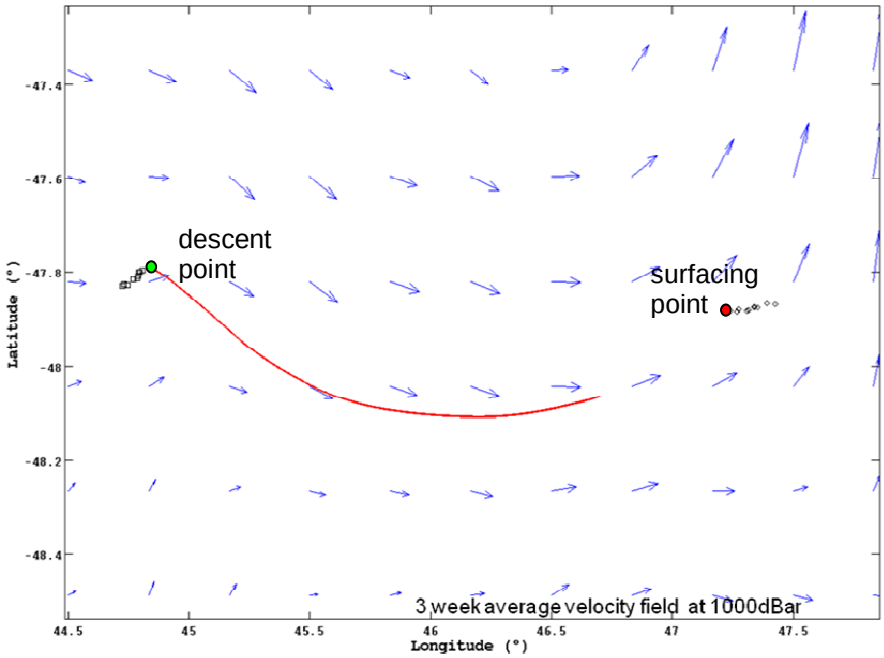


***Figure 7.2**: Example of a simulated Argo float trajectory in a relatively simple velocity field with moderate misfit in surfacing position. In this case the velocities in the SatGEM model underestimate the velocities implied by the Argo float.*

Figure 7.3 provides an example of a simulated trajectory with a large misfit in a region with a complex velocity field. Examination of the velocity vectors in this case suggest that multiple bifurcation or branch points are present in the velocity field and that in a complex field such as this the simulated trajectory may be highly sensitive to perturbations in the initial descent point and local variations in velocity structure. For the case shown in Figure 7.3 it is quite reasonable to infer from the velocity vectors that the simulated path of the float would have much more closely

replicated the actual surfacing point had the descent point been located slightly south of its actual position in the velocity field.

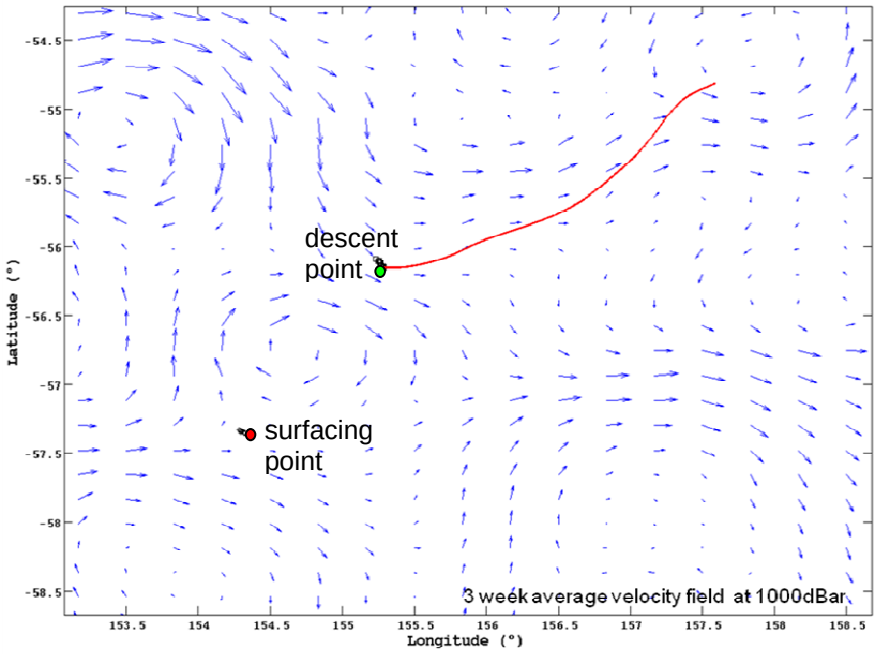


***Figure 7.3**: Example of a simulated Argo float trajectory in a complex velocity field which results in a large misfit in surfacing position. The velocity field in this case has multiple bifurcation and branch points.*

7.2 Geographic Distribution of Float Data

The spatial distribution of the density of usable Argo cycles is shown in Figure 7.4. The pattern in Figure 7.4 is affected by both the distribution of the Argo data and also constraints on the SatGEM model resulting from depth limitations.

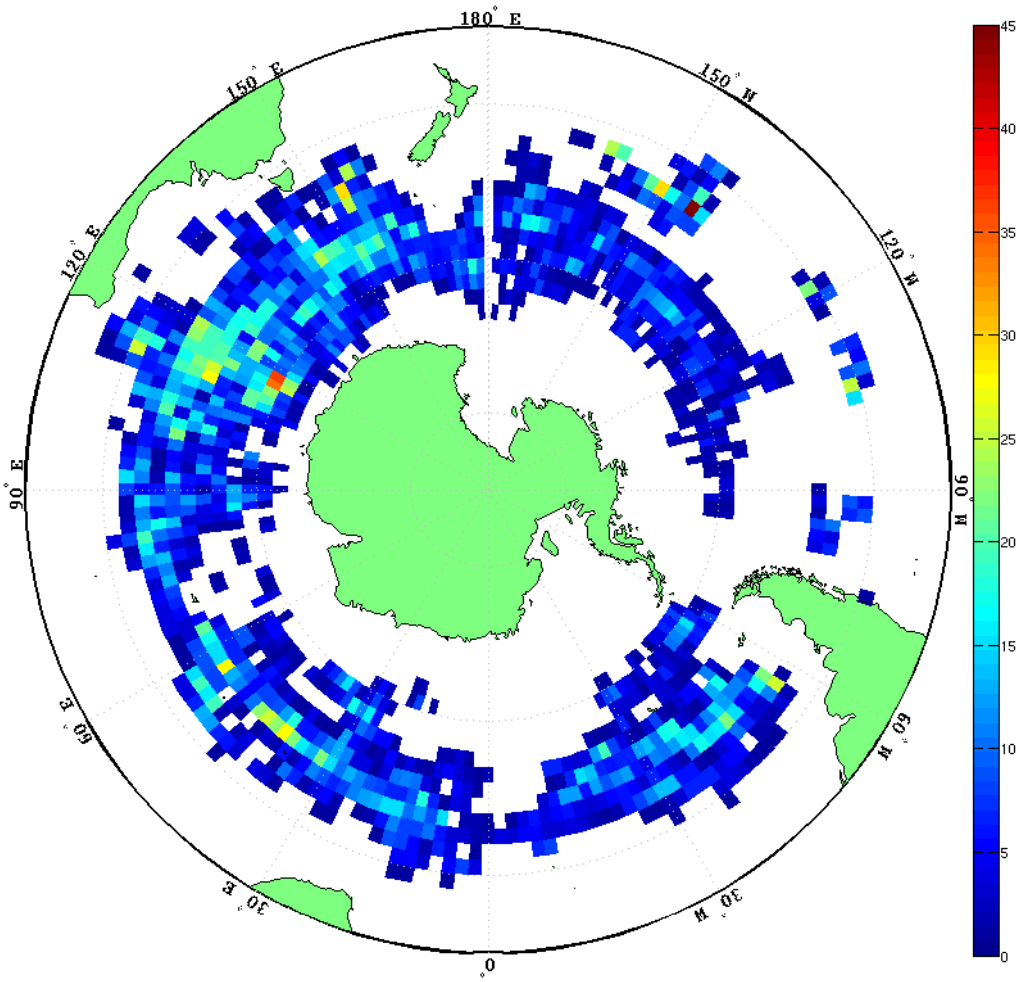


***Figure 7.4**: Circumpolar density of data points (by descent location) in 2° by 2° cells.*

Data density is highest in the ACC south of Australia. Of particular note is the lack of data in the Eastern Pacific, this is largely an artefact of the late start to Argo Float deployment in this region. Also significant are the narrow bands of float coverage in the vicinity of Kerguelen Island (~70°E) and in the Mid-Atlantic (~10°W). The Kerguelen anomaly corresponds broadly to the shallow water of the Kerguelen Plateau and the low data density region in the Atlantic also corresponds closely to a region of shallow water. These regions of no coverage result primarily from gaps in

the SatGEM fields due to the inability to resolve velocity variations in shallow water (<2000m) rather than due to a lack of Argo Float coverage in these areas.

7.3 Path Length and Misfit

Path Length

Figure 7.5 shows the simulated path length ($|P_{sim}|$) for all float cycles plotted as a function of actual path length ($|P_{argo}|$). There is a general but poorly defined trend of increasing $|P_{sim}|$ with increasing $|P_{argo}|$ but the R^2 value for a linear regression between $|P_{sim}|$ and $|P_{argo}|$ is only 0.28 and there are a significant number of outliers that largely correspond to paths in complex regions of the velocity fields.

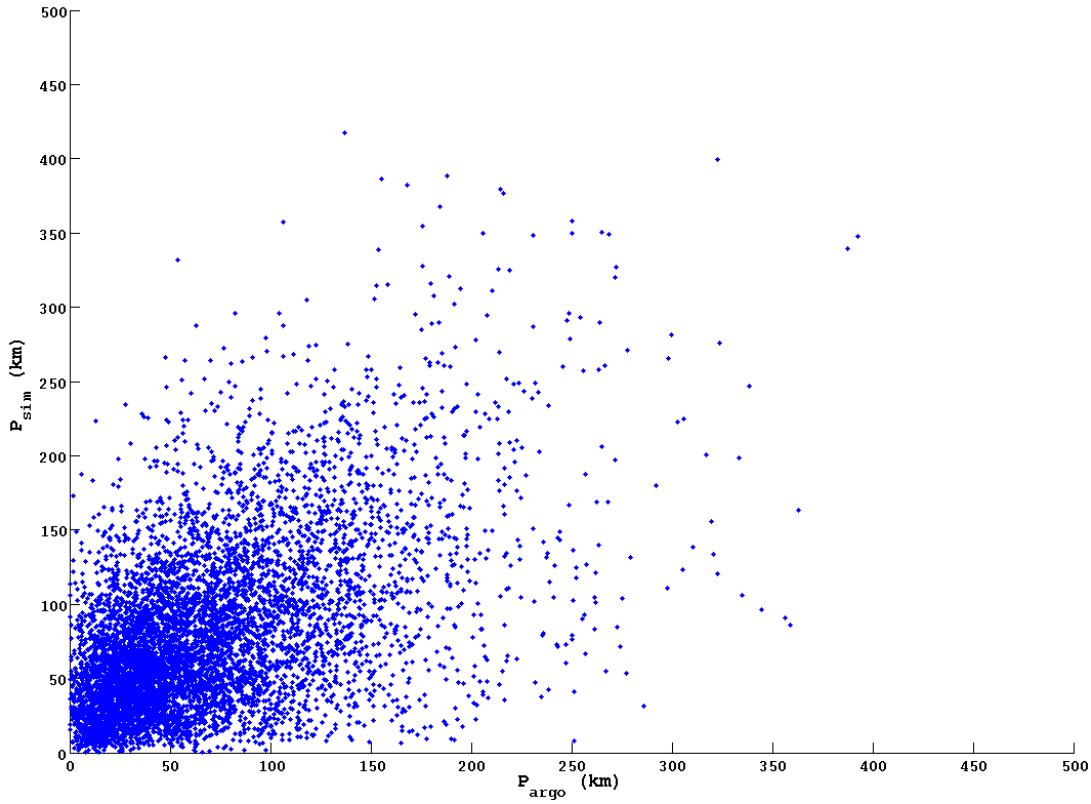


***Figure 7.5**: Scatter plot of $|P_{sim}|$ vs $|P_{Argo}|$.*

Geographic Distribution of Path Length

Absolute values of real (P_{argo}) and simulated (P_{sim}) path lengths were mapped onto a ¼° by ¼° grid using an inverse distance squared interpolator with a search radius of 1 degree (Figures 7.6 and 7.7). Trends and variations in the gridded dataset are more readily interpreted that images showing the distributions of data points and the gridding process acts as a low-pass filter to provide a more synoptic view of

generalised variations. The gridding process also produces a temporally averaged dataset since path lengths for the entire duration of the model are represented in the same image.

There is a reasonable degree of agreement between the actual and simulated path lengths regarding large-scale features such as the band of longer path lengths running from Kerguelen Island to the Campbell Plateau south of New Zealand and a similar feature in the South Pacific Ocean. These main high amplitude features broadly correspond to the time averaged position of the principle current flow within the ACC.

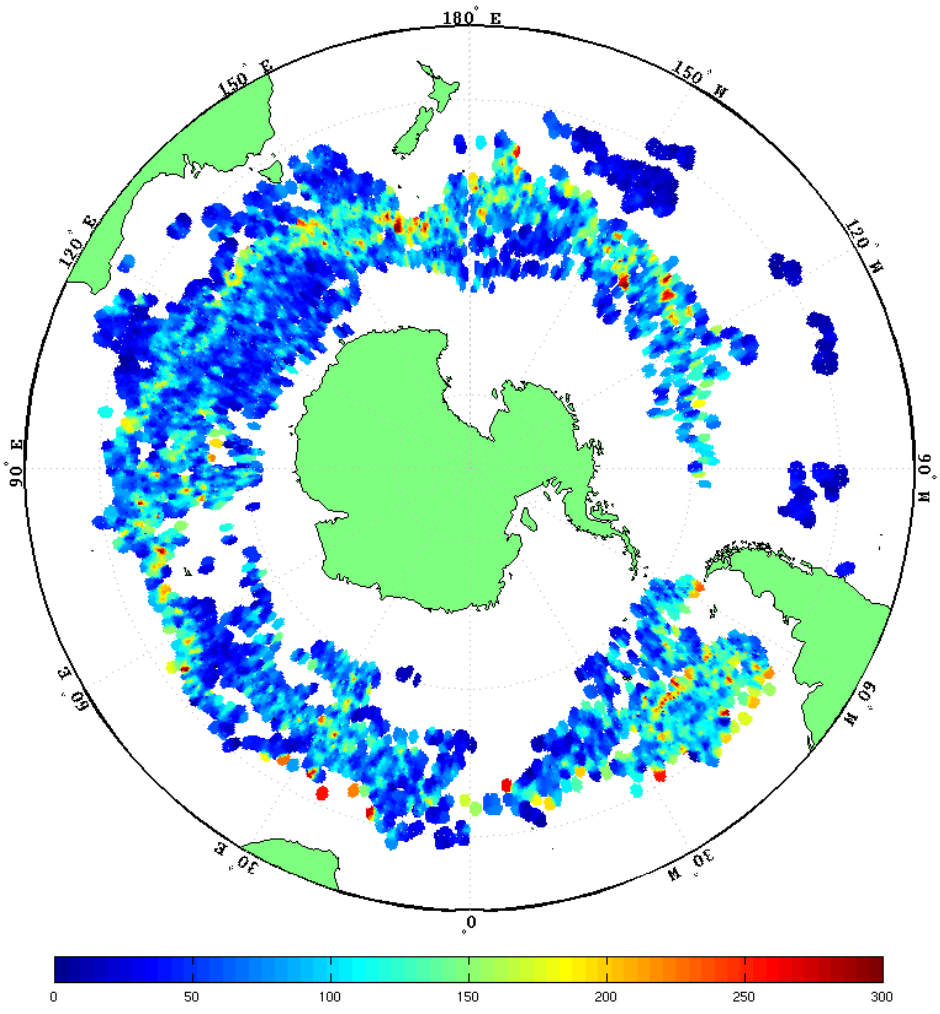


***Figure 7.6**: Argo path lengths (km) gridded on to a ¼° by ¼° grid.*

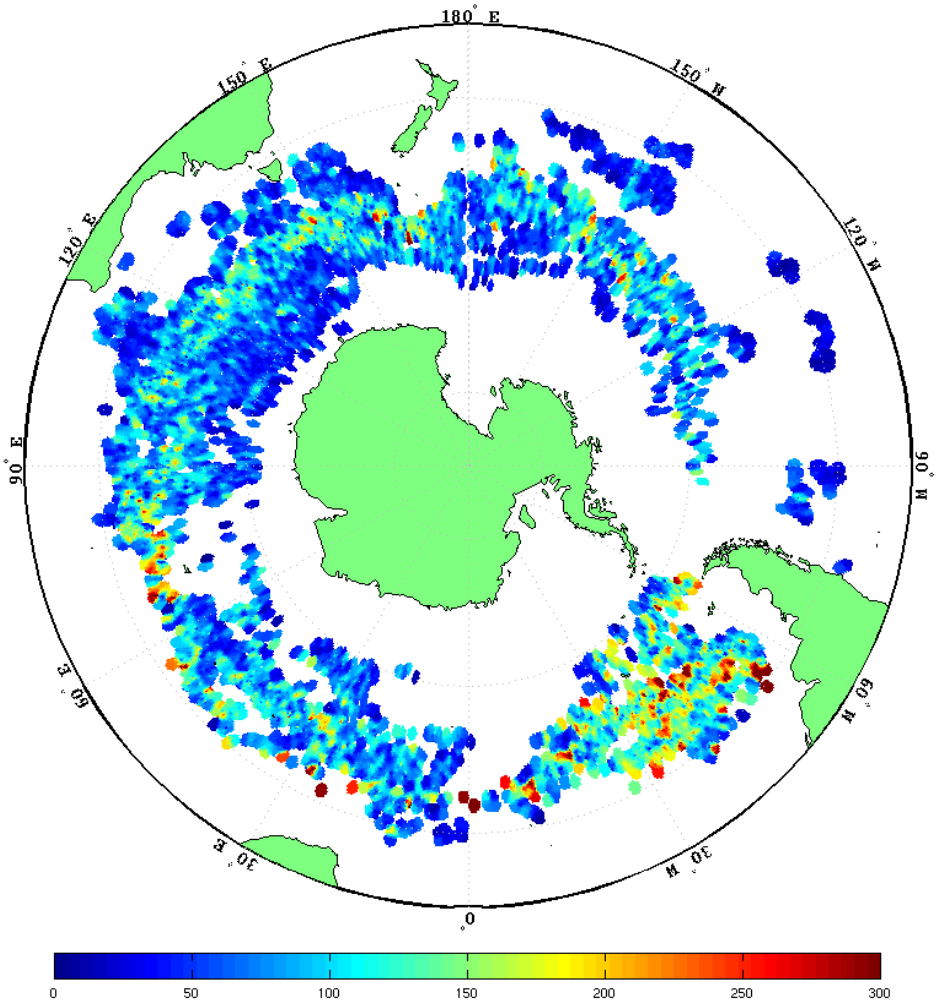


***Figure 7.7**: Simulated path lengths (km) gridded on to a ¼° by ¼° grid.*

Although the general form of the high amplitude features is broadly the same for the actual and simulated data, the simulated path length image shows portions of these zones with higher velocity than the actual path image. The two path length images show significant differences in amplitude for some features. This is most noticeable in the western Atlantic and just to the north of Kerguelen Island. In both cases the P_{sim} values are higher than the corresponding values for P_{argo}.

Misfit

Figure 7.8 shows the absolute value of the misfit between simulated and actual surfacing position plotted as a function of actual path length ($|P_{argo}|$). There is a general but poorly defined trend of increasing misfit with increasing path length but the R^2 value for a linear regression between misfit and $|P_{argo}|$ is only 0.21 and there are

a significant number of outliers with very large misfits even for relatively short path lengths.

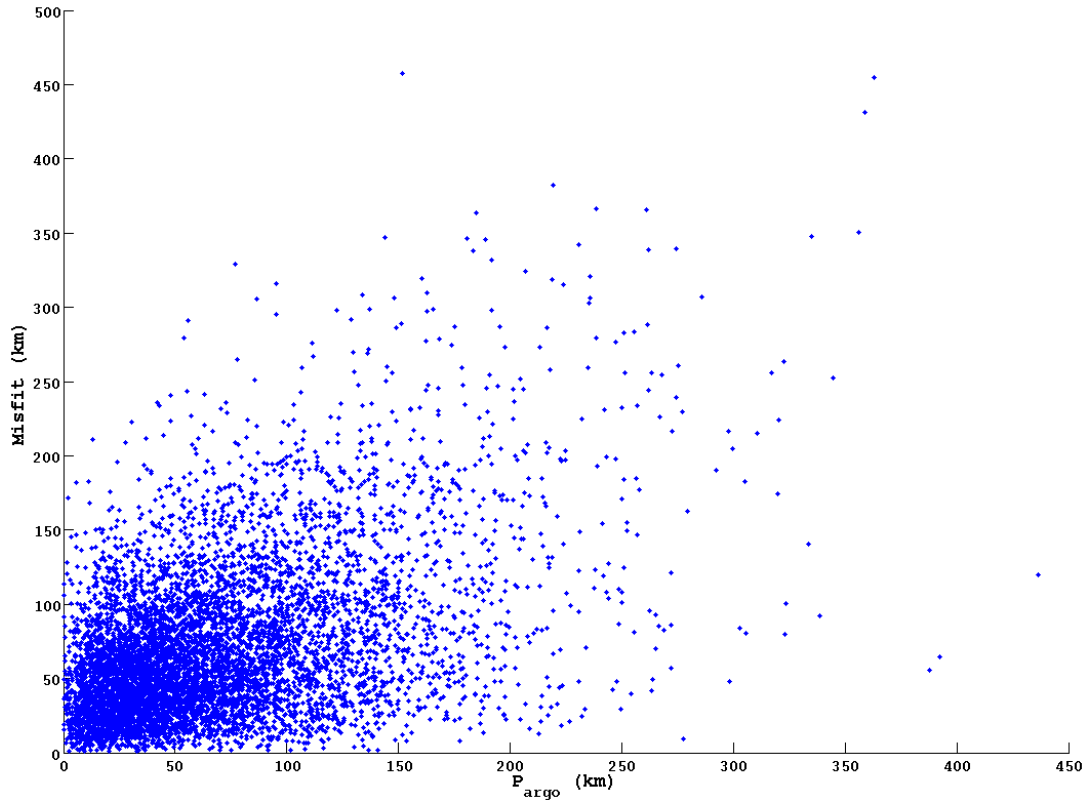


***Figure 7.8**: Scatter plot of misfit (ε) vs $|P_{Argo}|$.*

Relative Misfit

Figure 7.9 shows relative misfit plotted as a function of actual path length ($|P_{argo}|$). Large relative misfits are most common for short path lengths. This is most significant for path lengths of less than 50km and exceptionally short paths exhibit relative misfits of up to 120, indicating that the difference in simulated and Argo paths can be up to 120 times the Argo path length. This result is likely due to fact that the SatGEM velocity fields can only resolve features significantly larger than the grid cell size (1/3° by 1/3°). In regions with complex small-scale velocity features the actual Argo float may travel a convoluted path but the details of this motion are not adequately defined in the velocity fields and the numerical model may significantly overestimate average velocities and hence net displacements.

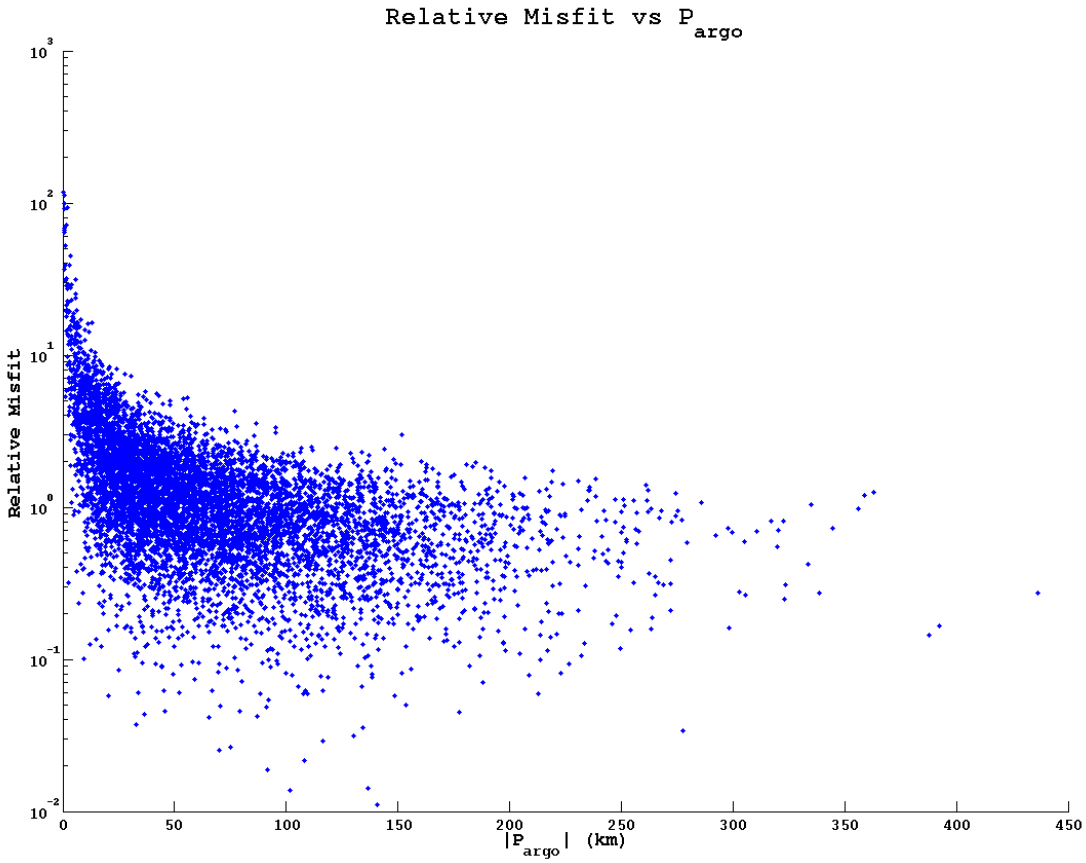


***Figure 7.9**: Scatter plot of ε_r vs $|P_{Argo}|$.*

Overall, the mean relative misfit is 1.80. The majority of the extreme misfits occur for very short path lengths and the mean relative misfit reduces to 0.91 when trajectories with path lengths of less than 50km are removed. As a result, for some elements of the subsequent analysis path lengths of less than 50km are not considered. The removal of Argo float cycles with actual path length ($|P_{argo}|$) less than 50km prior to data analysis can be justified due to the resolution of the SatGEM model (1/3°) which means that velocity features of at least twice these dimensions are not adequately resolved in the model. The 1/3° by 1/3° grid resolution of the SatGEM model corresponds to about 37km by 30km at 35°S and 37km by 16km at 65°S.

For the entire dataset 42% of cycles display large relative misfit (ε_r>1.25) and only 8% display small relative misfit (ε_r<0.3). Exclusion of Argo cycles with actual path lengths less than 50km reduced the percentage of cycles with large relative misfits to 24% and increased the percentage displaying small relative misfits to 12%. This suggests that approximately half of the cycles with large relative misfit may be the result of inadequate resolution in the SatGEM velocity fields. A qualitative

assessment based on the form of time averaged velocity fields in the vicinity of the float cycles with path length in excess of 50km and very large relative misfit suggests many of these occur in zones with complex flow and multiple bifurcation points.

Geographic Distribution of Misfit

Float cycles were sorted into three categories by relative misfit (low, ε_r <0.3; medium, 0.3≤ε_r ≤1.25; high, ε_r >1.25) before being mapped (Figure 7.10). Additionally both the absolute misfit and relative misfit were gridded with a ¼° by ¼° grid mesh size using an inverse distance squared interpolator with a search radius of 1 degree to better visualise geographic variability (Figures 7.11 and 7.12).

Figure 7.10 shows the distribution of float cycles grouped by misfit categories. This plot is too cluttered to easily pick out any patterns or relationships but has been included to illustrate the geographic distribution of the classes used in subsequent analysis. The grid of relative misfit (Figure 7.12) is generally easiest to interpret.

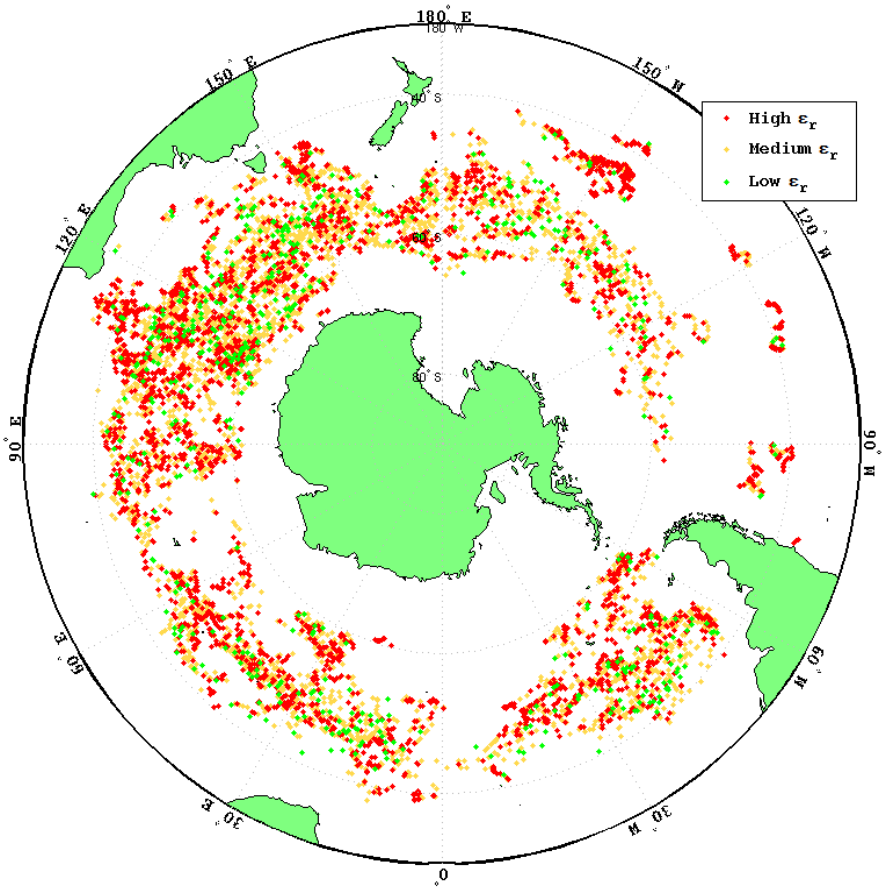


***Figure 7.10**: Circumpolar distribution of all float cycles sorted into three categories based on ε_r.*

For the majority of the study region the relative misfit is less than 1.5. With the exception of two patches near 110°E and 150°W, relative misfits of greater than 4.5 are irregularly distributed within areas of lower misfit. The distribution of high and low relative misfit cycles is examined in more detail in section 7.5.

Figure 7.11 shows the 'raw' misfit in km. Of note is the region of particularly large misfit east of near the Campbell Plateau (170°E). Other areas of elevated misfit are observed south of Tasmania; near Kerguelen Island; in the western Atlantic and in the Pacific sector between 120 and 150°W. The likely explanations for these areas of misfit are discussed in sections 7.5 and 7.6. The patch of very large relative misfits near 150°W (Figure 7.12) corresponds with a region in which the absolute misfits and both Argo and simulated path lengths are small.

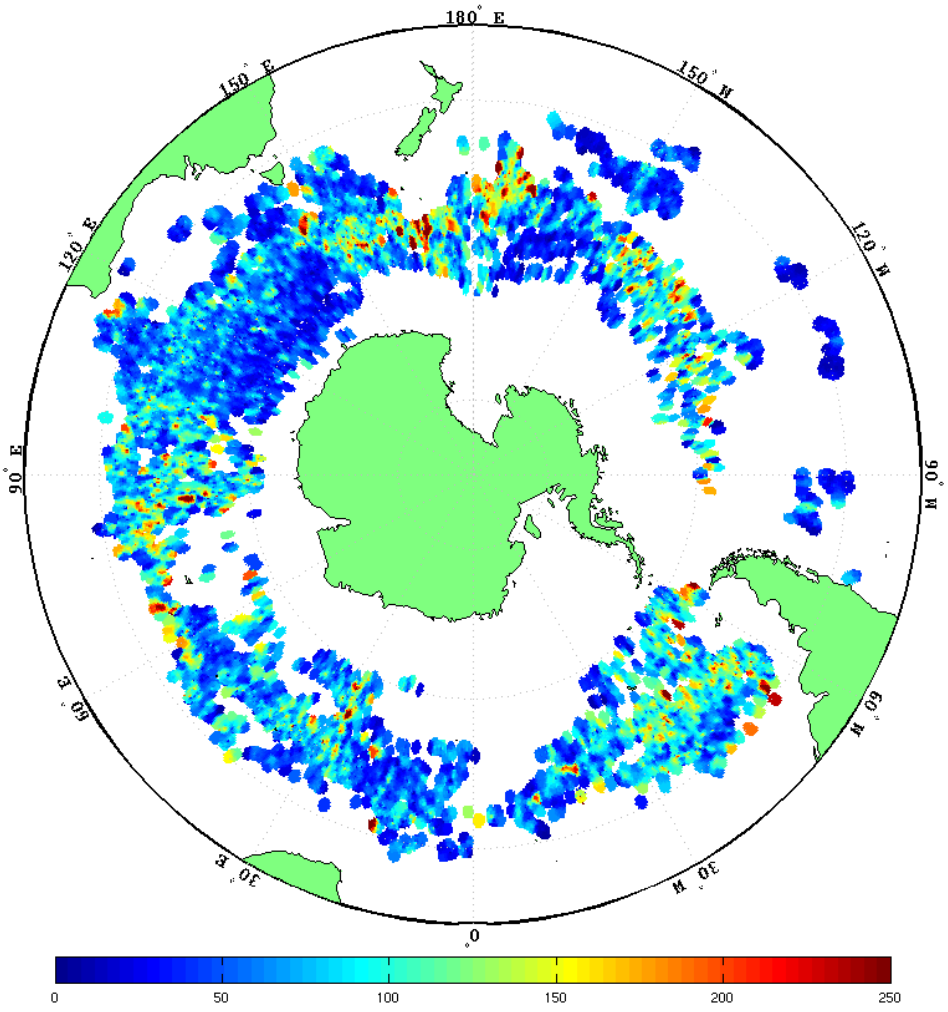


***Figure 7.11**: Misfits (km) gridded on to a ¼° by ¼° grid. Misfits greater than 250km are truncated.*

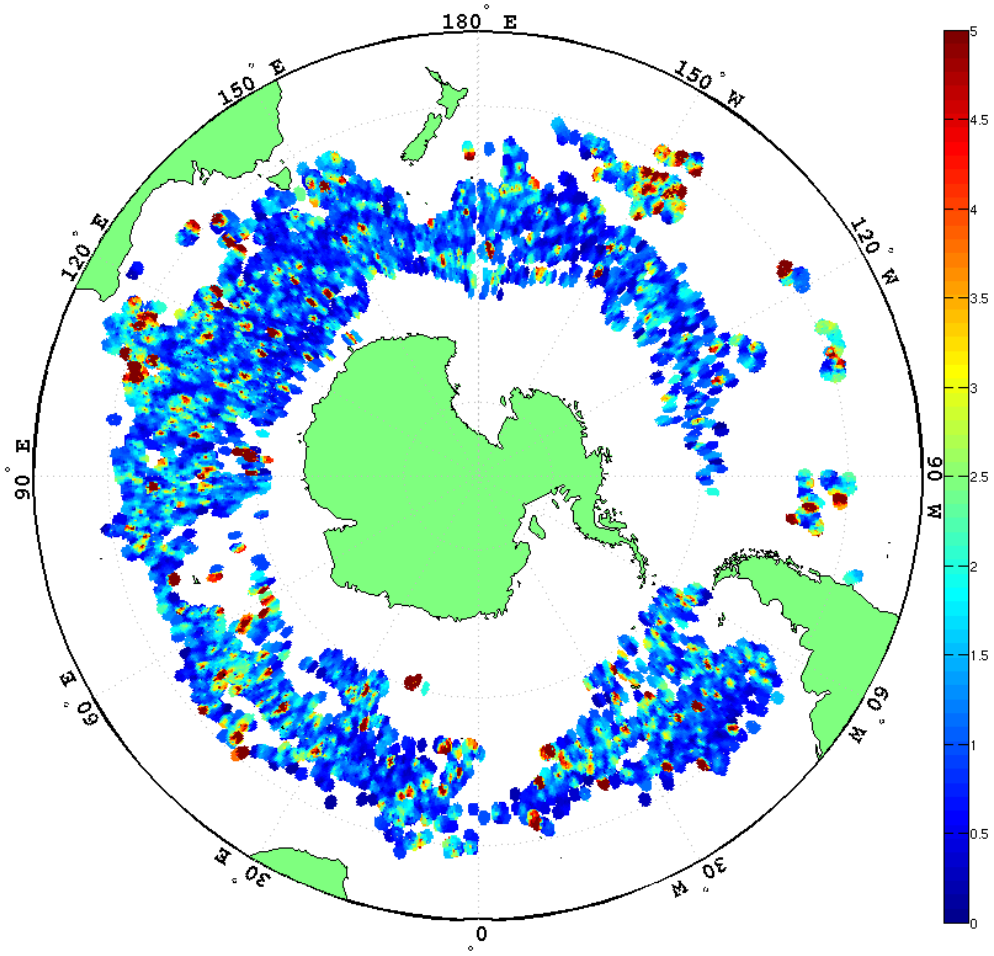


***Figure 7.12**: Relative misfits gridded on to a ¼° by ¼° grid. Values greater than 5 are truncated.*

7.4 Histograms and Statistics

The overall statistical performance of the SatGEM velocity fields when compared to Argo float trajectories can be assessed by analysis of distributions of path lengths and data misfits.

ε, P_{sim} and P_{Argo}

The histograms of latitudinal and longitudinal components of Argo path length (P_{argo}) are shown in Figures 7.13a and 7.13b. The latitudinal component is approximately normally distributed with a mean close to zero. The zero mean in this case is indicative of the fact that on a global, depth integrated basis there is no net water flow in a north-south direction across the Southern Ocean. The histogram of the longitudinal component of Argo path length (P_{argo}) is skewed towards positive (eastwards) displacements.

The mean longitudinal displacement is ~26km east and this reflects the overall influence of the ACC in the west to east transport of water in the Southern Ocean.

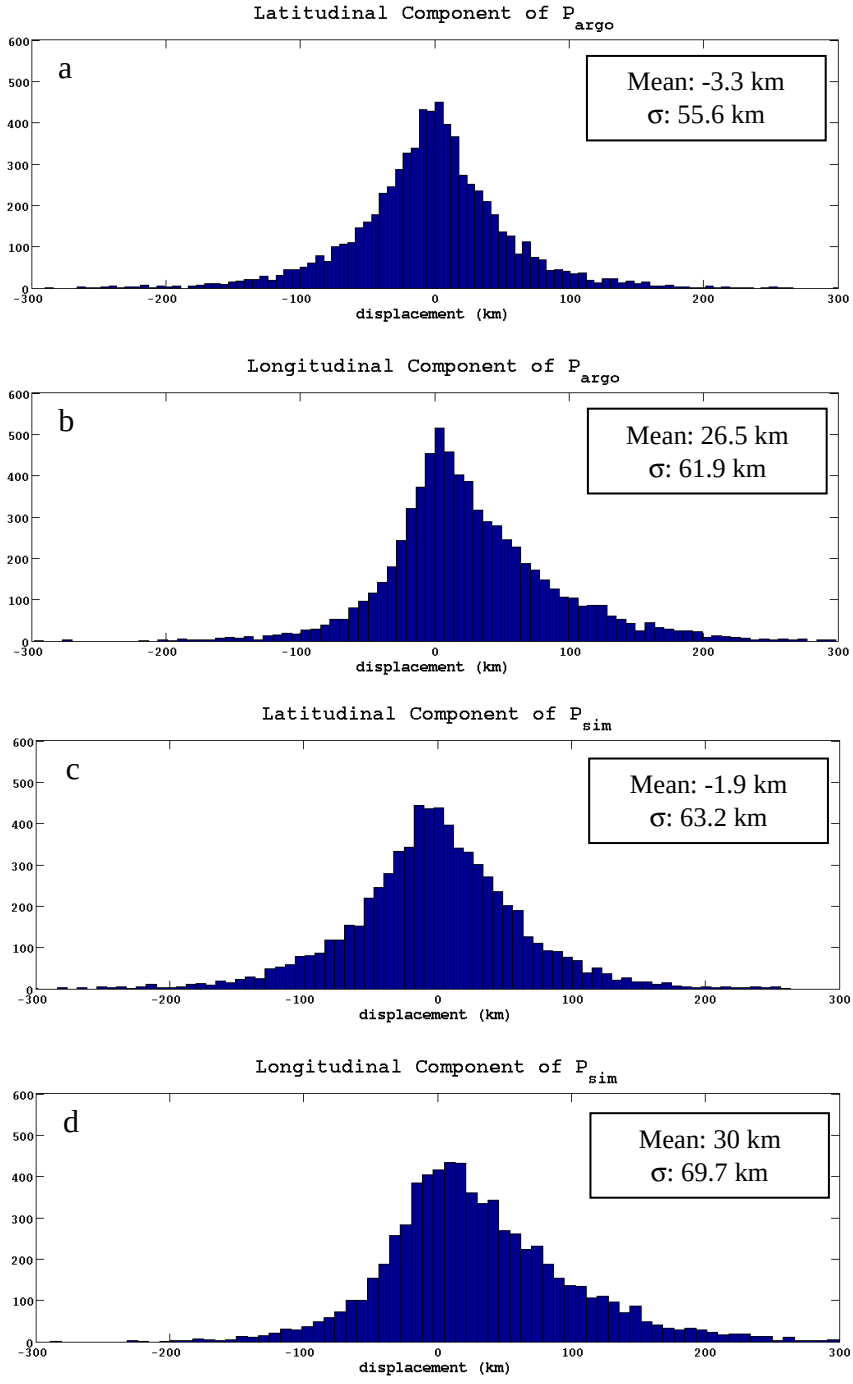


***Figure 7.13**: Histograms of: (a) latitudinal component of Argo path (P_{argo}), (b) longitudinal component of Argo path t($P_{argo,}$), (c) latitudinal component of simulated path (P_{sim}),(d) longitudinal component of simulated path (P_{sim}). Positive numbers indicate north or east, negative numbers denote south or west.*

The histograms of simulated path length (P_{sim}) are shown in Figures 7.13c and 7.13d. The latitudinal component, like P_{argo}, is also normally distributed with mean located close to zero. The histogram of longitudinal displacements displays a similar eastward skewed distribution to the corresponding plot for P_{argo}. Although the general form of these two distributions are quite similar there are also some differences; the distributions of simulated path length although having similar standard deviations have slightly less well-defined peak values than the distributions for actual path length.

Overall, the distributions of simulated path length and Argo path length match reasonably closely suggesting that at least in an overall statistical sense the SatGEM velocity fields capture the gross flow features of the Southern Ocean.

The latitudinal and longitudinal histograms of misfit are shown in Figure 7.14. Both display a similar degree of spread and both are centred at or close to zero, suggesting that the SatGEM model shows little overall directional bias. The means and standard deviations of the path length data (Figure 7.13) confirm the observations.

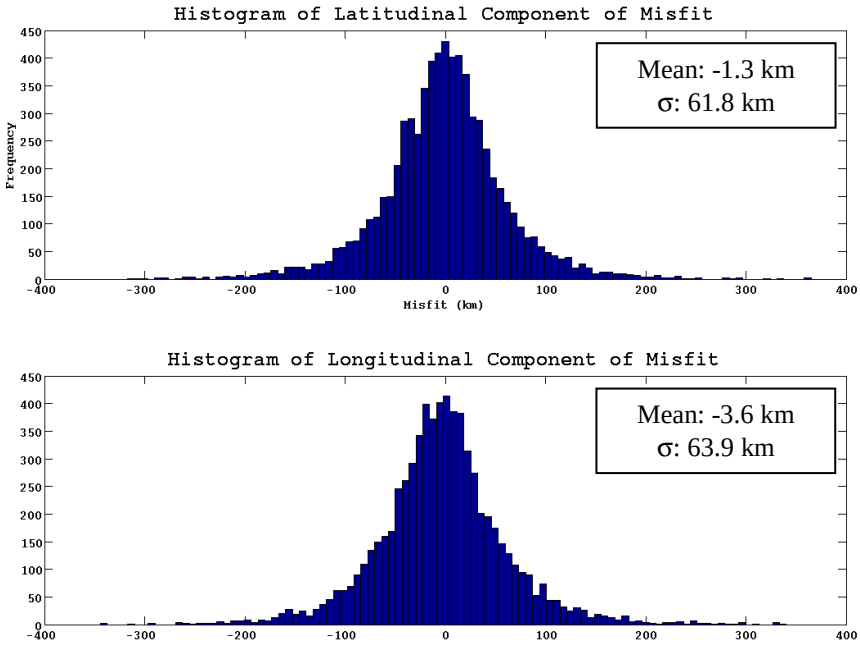


***Figure 7.14**: Histograms of latitudinal (top) and longitudinal (bottom) components of misfits. Direction conventions are as per Figure 7.13.*

The means and standard deviations of simulated and real path lengths are of similar size: the latitudinal means are within 2km of each other while the longitudinal

mean are separated by 3.5km. For both the latitudinal and longitudinal components the standard deviation is slightly larger for the simulated paths than the Argo paths. Misfit standard deviations are of similar magnitude to the standard deviations of both real and simulated paths. Also both latitudinal and longitudinal misfit standard deviations are within 2km of each other. The misfit distribution further reinforces the suggestion that on the whole the SatGEM model displays little directional bias.

Misfit Parallel and Perpendicular to SSH

The histograms for latitudinal, longitudinal, across SSH contour and along SSH contour misfits are generally similar (Figure 7.15) for floats with Argo path length greater than 50 km.

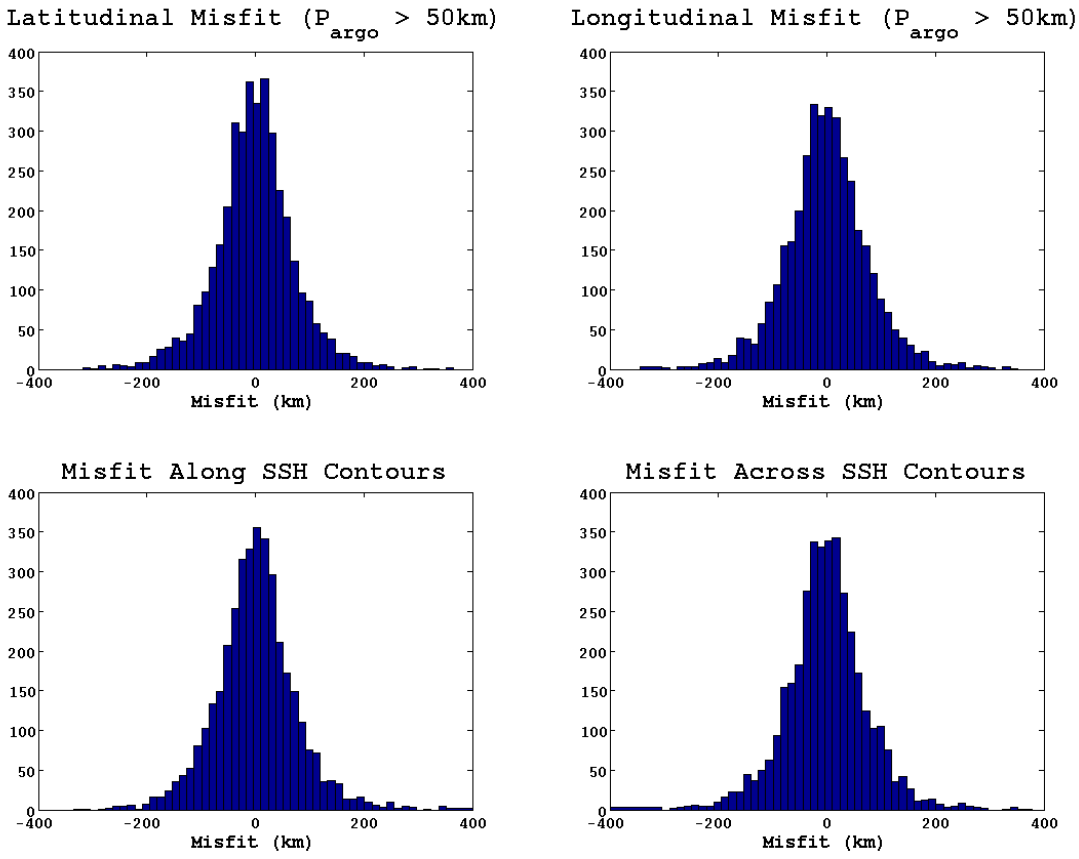


Figure 7.15*: Histograms of misfit components: latitude (top left), longitude (top right), along (bottom left) and across (bottom right) SSH contours. All histograms are for* P_{argo}*>50km.*

Additionally, means and standard deviations vary by less than 3km in all cases. These results further reinforce the notion that the SatGEM model displays little directional

bias in velocity and also suggests that the performance of the model is not sensitive to the orientation of SSH contours.

Angle between P_{sim} and P_{Argo} (θ)

The angle between P_{sim} and P_{argo} for the entire dataset (Figure 7.16a) is a symmetrical but not normally distributed variable. The distribution retains its basic form upon removal of short path lengths (Figure 7.16b). However examination of the angular discrepancy on the basis of relative misfit shows that the overall distribution can be considered as the sum of two populations. The distributions of large relative misfits are approximately uniform (Figures 7.16c and 7.16d) while the distribution of small and medium relative misfits follows a normal distribution (Figures 7.16e and 7.16f) for all data and for Argo path lengths in excess of 50 km. For the medium and small misfit subset the means are very close to zero degrees.

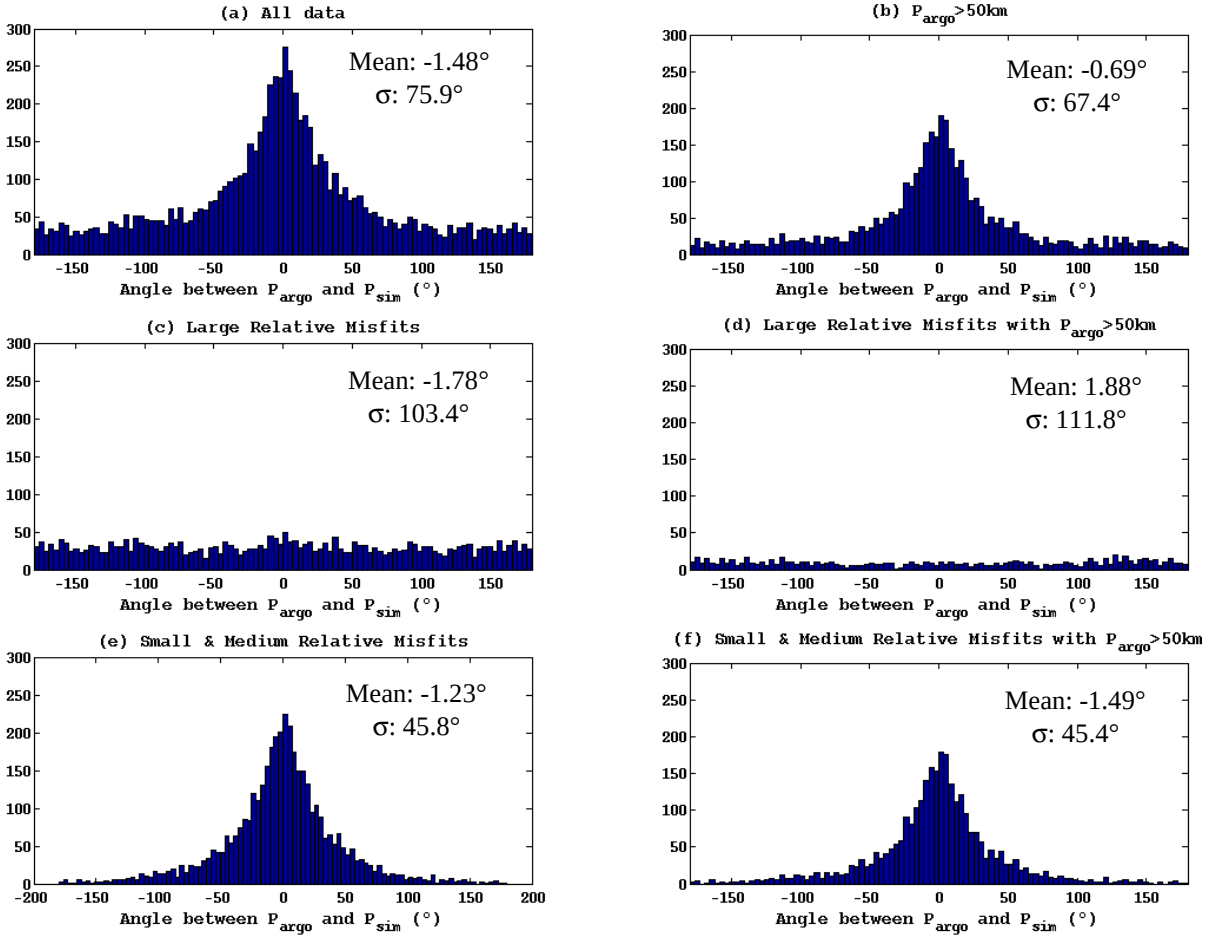


***Figure 7.16**: Histogram of the angle between P_{argo} and P_{sim} for: (a) all data, (b) P_{argo}>50km, (c) all ε_r >1.25, (d) all ε_r >1.25 and P_{argo}>50km, (e)ε_r <1.25, (f) ε_r <1.25 and P_{argo}>50km.*

The fact that the angular misfit is normally distributed for small and medium misfits suggests that for these paths there is simple statistical variability in the SatGEM velocity field relative to the actual field as recorded by the Argo data

The uniform distribution of angular misfit for the high relative misfit paths is suggestive of a more random process which may be related to the sensitivity to initial descent position in regions of complex velocity structure with many bifurcations and branch points.

7.5 Correlation Coefficients and R^2 Values

Entire Population

The correlation coefficient between $|P_{Sim}|$ and $|P_{Argo}|$ for the entire population of data was 0.53, indicative of a moderate level of correlation. This figure implies that the SatGEM model explains 28% of total variance in path length. The relationship for longitudinal components is similar with a correlation coefficient of 0.53 and approximately 28% of variance explained while Latitudinal performance is worse with a correlation coefficient of 0.465 (22% of the variance). Both of these values are marginally lower than the results obtained by Meijers et al. (2009a). The differences between the results obtained during this study and those obtained by Meijers et al. (2009a) are likely a function of the both the different techniques used to analyse the dataset and the range of floats used.

Geographic Distribution

Path length data were binned into 5° x 5° cells and the correlation coefficient calculated for data in each cell. The circumpolar distribution of R^2 values computed for $|P_{sim}|$ and $|P_{Argo}|$ is shown in Figure 7.17. R^2 values were only computed for cells containing five or more floats.

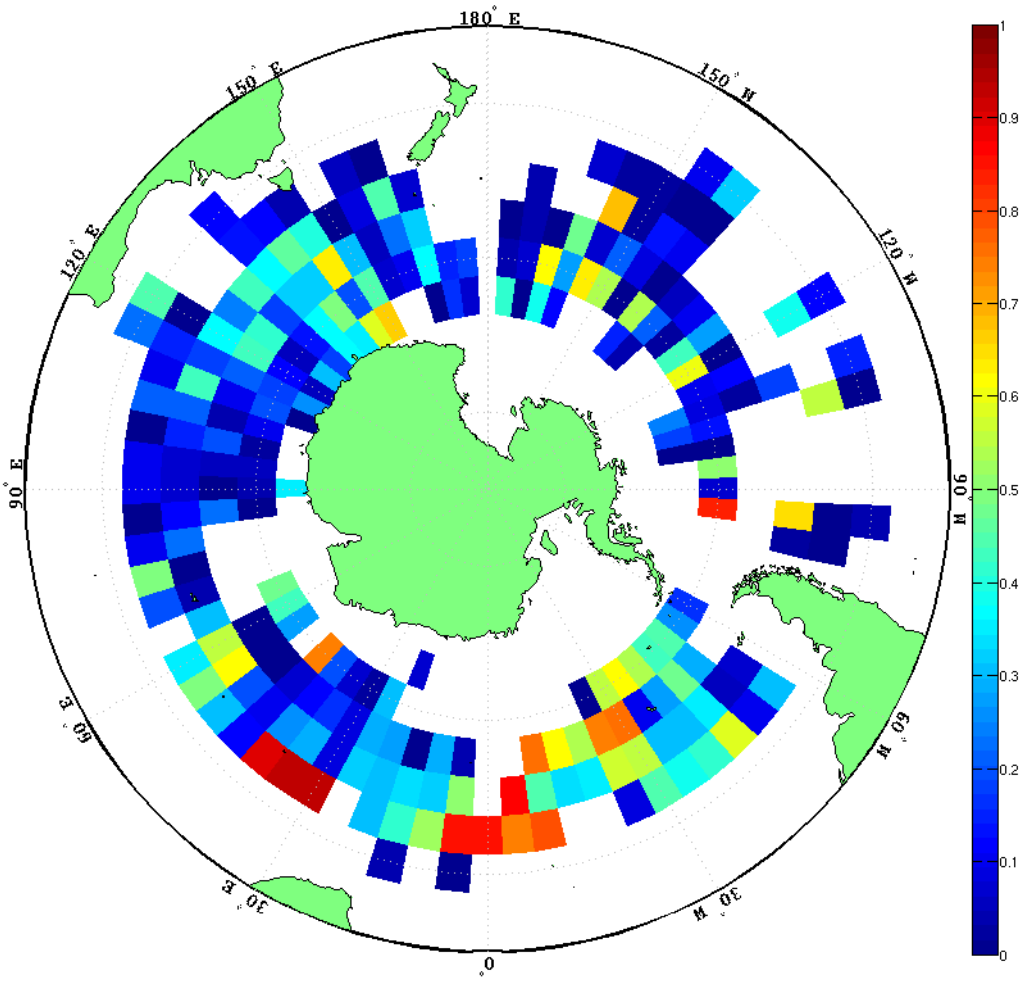


***Figure 7.17**: R^2 between $|P_{sim}|$ and $|P_{Argo}|$ for 5° x 5° cells. R^2 values were only plotted for cells with 5 or more float cycles.*

Regions of exceptionally high R^2 values (typically in excess of 0.8) generally tend to coincide with areas of low data density and so cannot necessarily be regarded as reliable indicators of SatGEM performance in those regions. Areas of moderate R^2 values (around 0.3 to 0.7) are seen South of Australia and in the Western Atlantic. Other notable features include the zones to the west of Kerguelen Island and New Zealand which display lower R^2 values than adjacent regions. The area directly adjacent to the Brazil-Falklands Confluence displays reduced performance relative to the rest of the Atlantic zone.

R^2 values for the latitudinal and longitudinal components are shown in Figure 7.18. Generally these maps demonstrate similar features to the R^2 maps for absolute path length.

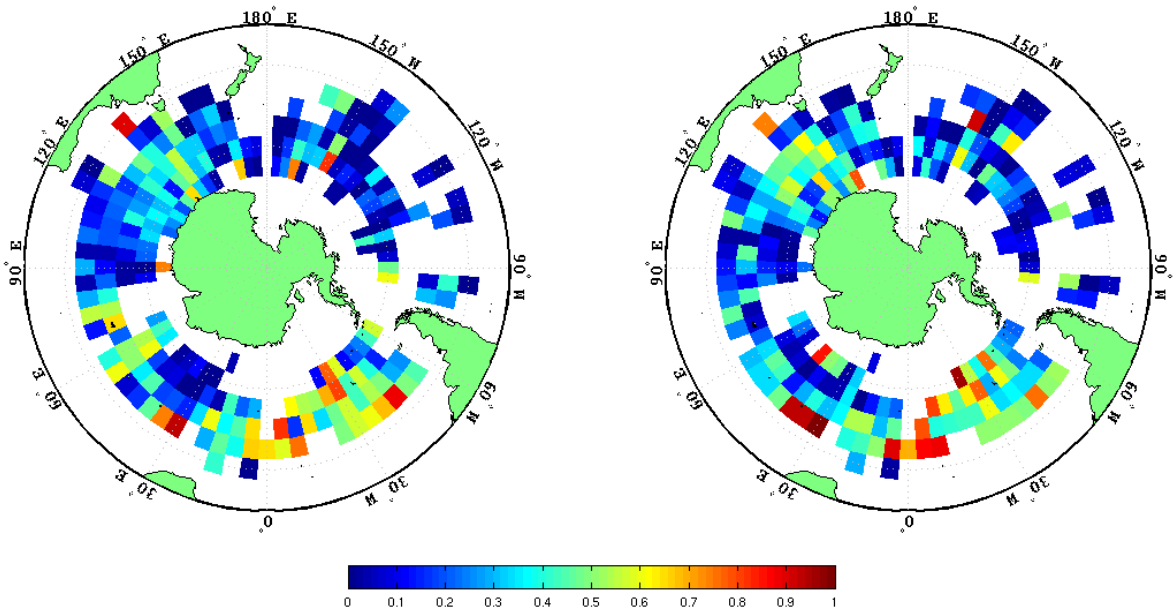


***Figure 7.18**: R^2 for Latitudinal (left) and Longitudinal (right) components of P_{sim} and P_{Argo} for 5° x 5° cells. R^2 values are only plotted for cells with 5 or more float cycles..*

However there is one significant exception, both the latitudinal and longitudinal components display higher R^2 values near the Brazil-Falklands Confluence than the R^2 for absolute path lengths.

Regional Analysis

Correlation Coefficients and the fraction variance captured for 30° wide segments of the Ocean and the ACC are shown Table 7.1. The percentage variance in absolute path length captured by the SatGEM dataset varies between 13% and 50%. The regions of 60°-90°E and 150°-180°W display reduced overall R^2 values relative to adjacent regions. The reduced performance of these two regions is not unexpected as the former coincides with the Kerguelen Plateau and the latter coincides with the Campbell Plateau. Both of these features are associated with regions of high eddy activity (Phillips and Rintoul 2000; Rintoul et al. 2001; Meijers et al. 2009a) and thus complex velocity fields. The region of 30-60°W, which borders the Brazil-Falklands Confluence also displays reduced performance relative to adjacent regions. This is not unexpected as the Brazil-Falklands Confluence is a known area of weakness for GEM-based techniques (Sun and Watts 2001).

Region	180 - 150° W	150 - 120° W	120 - 90° W	90 - 60° W	60 - 30° W	30 - 0° W	0 - 30° E	30 - 60° E	60 - 90° E	90 - 120° E	120 - 150° E	150 - 180° E	ACC
Tot. R	0.46	0.56	0.59	0.71	0.53	0.68	0.52	0.48	0.36	0.48	0.58	0.43	0.51
Tot. R^2	0.21	0.31	0.35	0.50	0.28	0.46	0.27	0.23	0.13	0.24	0.34	0.18	0.26
Lat. R	0.27	0.20	0.14	0.48	0.65	0.73	0.46	0.49	0.38	0.41	0.57	0.36	0.49
Lat. R^2	0.08	0.04	0.02	0.23	0.43	0.53	0.21	0.52	0.15	0.17	0.32	0.13	0.24
Lon. R	0.40	0.48	0.28	0.40	0.68	0.69	0.52	0.50	0.43	0.48	0.65	0.39	0.54
Lon R^2	0.16	0.23	0.08	0.16	0.46	0.48	0.27	0.25	0.19	0.23	0.42	0.16	0.30

***Table 7.1**: Correlation coefficients and R^2 values for 30° longitudinal segments of the Southern Ocean and for the ACC (SSH of 0.7m to 1.6m, Southern branch of sACCF and Northern branch of SAF respectively)*

The latitudinal path length correlation coefficients vary widely between 0.14 (90°-120°W) and 0.73 (0°-30°W), with the correlation coefficient of 0.14 being a significant outlier, the next smallest is 0.21 (for the region 120°-150°W). The range of correlation coefficients is wider than that obtained by Meijers et al. (2009a), this is likely a result of the numerical model's sensitivity to initial conditions. The percentage of latitudinal variance captured (R^2 value) by the SatGEM velocity fields also varies considerably, from 2% in the region of 90-120°W and 53% in the region of 0-30°W. All of the particularly poor R^2 values occur in the zone 90-180°W. The western limit of this region lies just downstream of the Campbell Plateau and as such some of the poor performance in the region of 150-180°W may be the result of eddy activity. Another factor which may contribute to the reduced performance both in this region and in the Pacific sector as a whole is the abnormally high number of float cycles found north of the Subantarctic front and thus outside the region that the SatGEM model is optimised for. For example, in the region 120-150°W 50% of cycles (250 cycles out of 502) are north of the STF, this number is also high, 48%(96 out of 200) for the region 90-120°W. These values compare to 25% (1751 out of 6984) for the entire dataset. Another contributing factor may be the lower number of floats active in the Pacific sector during the period covered by the SatGEM fields. The region around Kerguelen Island displays reduced R^2 values compared to the adjacent regions, though these values are still significantly higher than values found in the Pacific sector.

Longitudinal correlation coefficients range between 0.28 and 0.70. The 0.28 value is notably lower than all other values and also occurs for the same region that gave a particularly low latitudinal correlation coefficient (90° – 120° W). Excluding that value gives a range of 0.40 to 0.70 which is wider than but otherwise consistent with the results obtained by Meijers et al. (2009a). The longitudinal R^2 values indicate that the SatGEM captures between 7% (or 16% if the outlying value is neglected) and 48% of the variance in path length. In common with the latitudinal correlation variation, overall values local minima are most commonly associated with regions of known eddy activity. Like the latitudinal component, the longitudinal component correlation is generally worse performance for the Pacific Ocean compared to the rest of the circumpolar range, but this is not to the same extent as seen with the latitudinal component. For example, the local minimum R^2 associated with the Campbell Plateau is clearly visible (Figure 7.17).

7.6 Frontal Features and SSH

Mean Frontal Position

The circumpolar distribution of float cycles with $\varepsilon_r<0.3$ and $P_{Argo}>50$km is shown in Figure 7.19a. The cut off of $P_{Argo}>50$km was imposed to both improve the clarity of the diagrams and to filter out trajectories covering distances of less than or little more than the model resolution. The number of cycles within this category is unfortunately too limited to say much about the overall characteristics of the field beyond that there appears to be a slightly greater tendency for cycles to display good performance South of the SAF although this is a very poorly delineated trend.

The distribution of cycles with a relative misfit in excess of 1.25 is shown in Figure 7.19b. This figure, in conjunction with the plot of the ratio of poorly performing cycles to total cycles on a per cell basis (Figure 7.20) is more informative.

The proportion of large misfit cycles downstream (east) of the Kerguelen Plateau is significantly higher than in the zone immediately upstream. A similar if less pronounced effect is also seen in the region of the Campbell Plateau, an area of known eddy activity (Rintoul et al. 2001). An increased density of poorly performing floats is also observed near the Brazil-Falklands Confluence, a region where GEM performance is known to be poor (Sun and Watts 2001).

a.

b.

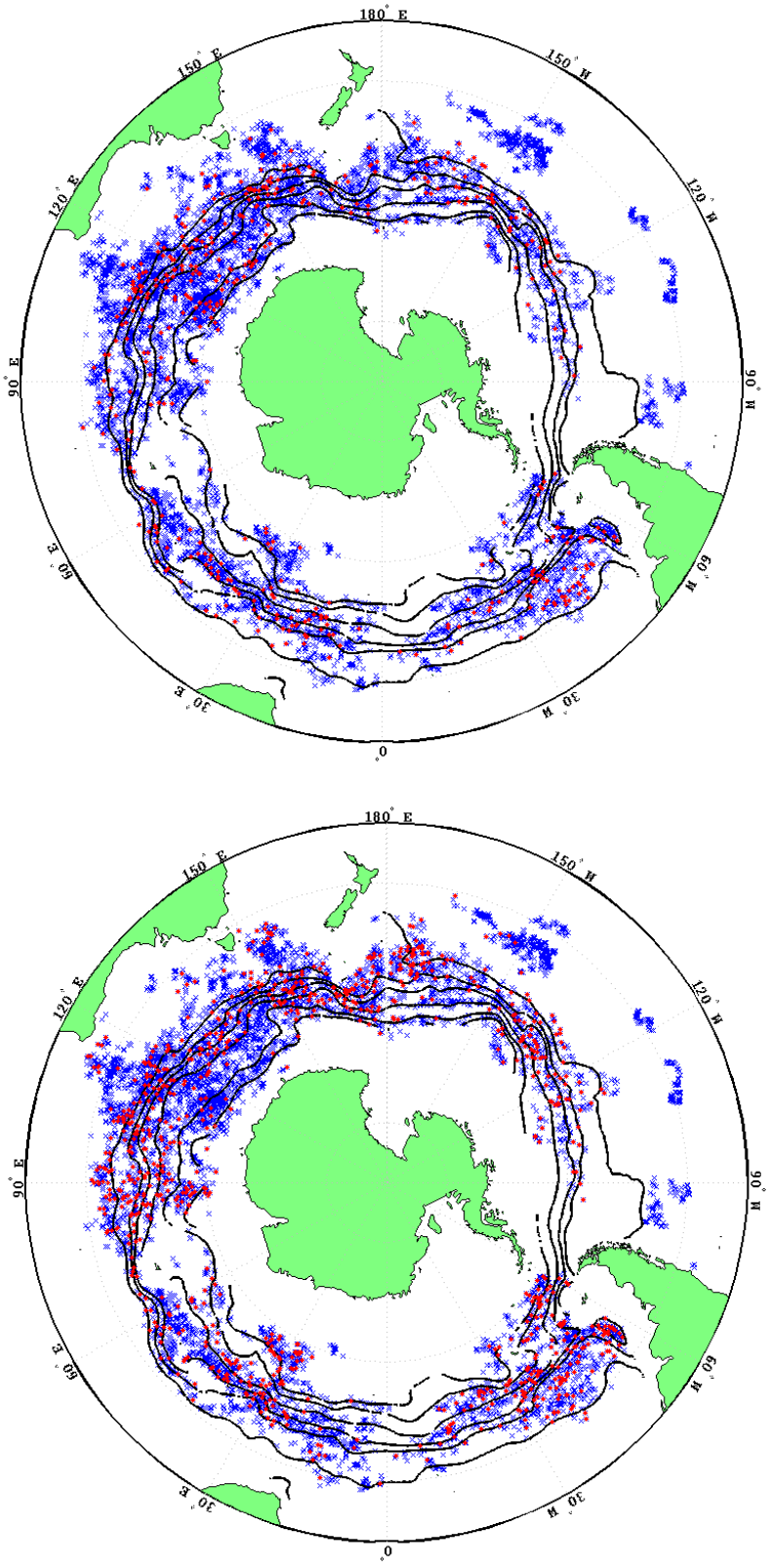


***Figure 7.19**: Distribution of float cycles with: (a) ε_r<0.3 and P_{Argo}>50km, (b) ε_r>1.25 and P_{Argo}>50km. Selected float cycles are highlighted in red. Mean frontal positions as defined by SSH are superimposed (see Section 6.5).*

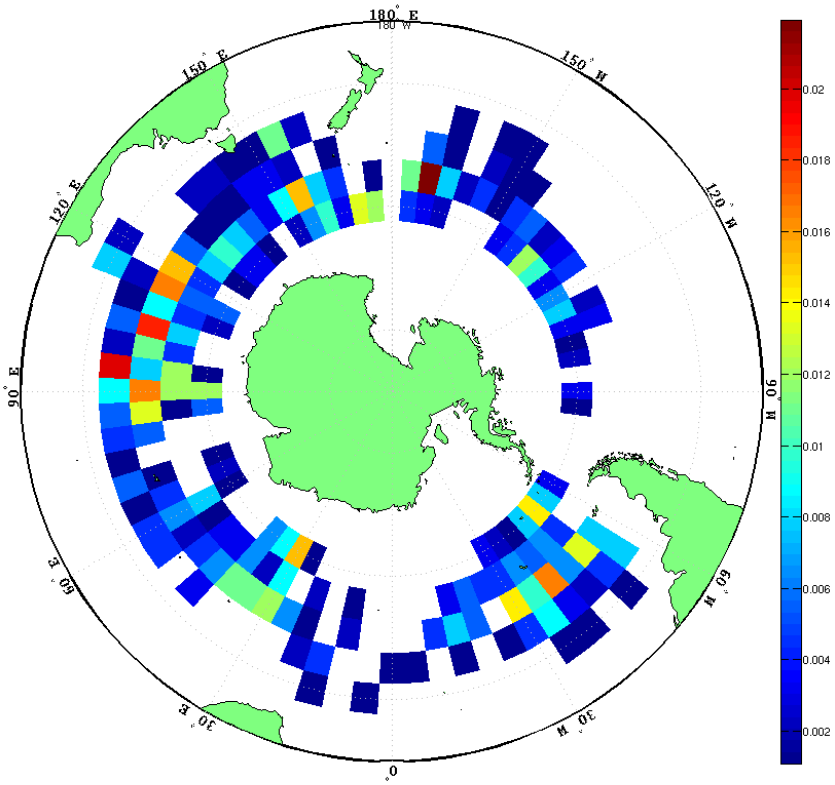


***Figure 7.20**: Ratio of float cycles with ε_r>1.25 and P_{Argo}>50km to all float cycles on a cell by cell basis for a 5° x 5° grid.*

Comparison with the circumpolar grid of R^2 values (Figure 7.17) shows that areas displaying a higher than typical concentration of cycles with large ε_r tend to correspond to regions of low R^2 between $|P_{argo}|$ and $|P_{sim}|$ relative to adjacent regions.

The longitudinal and latitudinal distribution of both low misfit (ε_r<0.3) and high misfit (ε_r>1.25) float cycles were also tested using the non-parametric Kolmogorov-Smirnov (KS) Test (Swan and Sandilands, 1995). The KS test suggests that the longitudinal distribution of low misfit cycles within the overall population can be considered as essentially random. However for high misfit cycles the distribution is not entirely random with higher proportions of high misfit cycles than would be expected from random sampling in the region 90°E to 120°E and to a less extent from 10°W to 40°W. These two regions also correspond to zones of low R^2 between $|P_{argo}|$ and $|P_{sim}|$ (Figure 7.17). The distribution of both low and high misfit cycles with respect to latitude can be regarded as essentially random based on the KS test.

Sea Surface Height

Scatter plots of relative misfit vs SSH for both the entire dataset and for float cycles with P_{Argo}>50km are shown in Figure 7.21. There is no obvious relationship between relative misfit and SSH apparent in the unfiltered data.

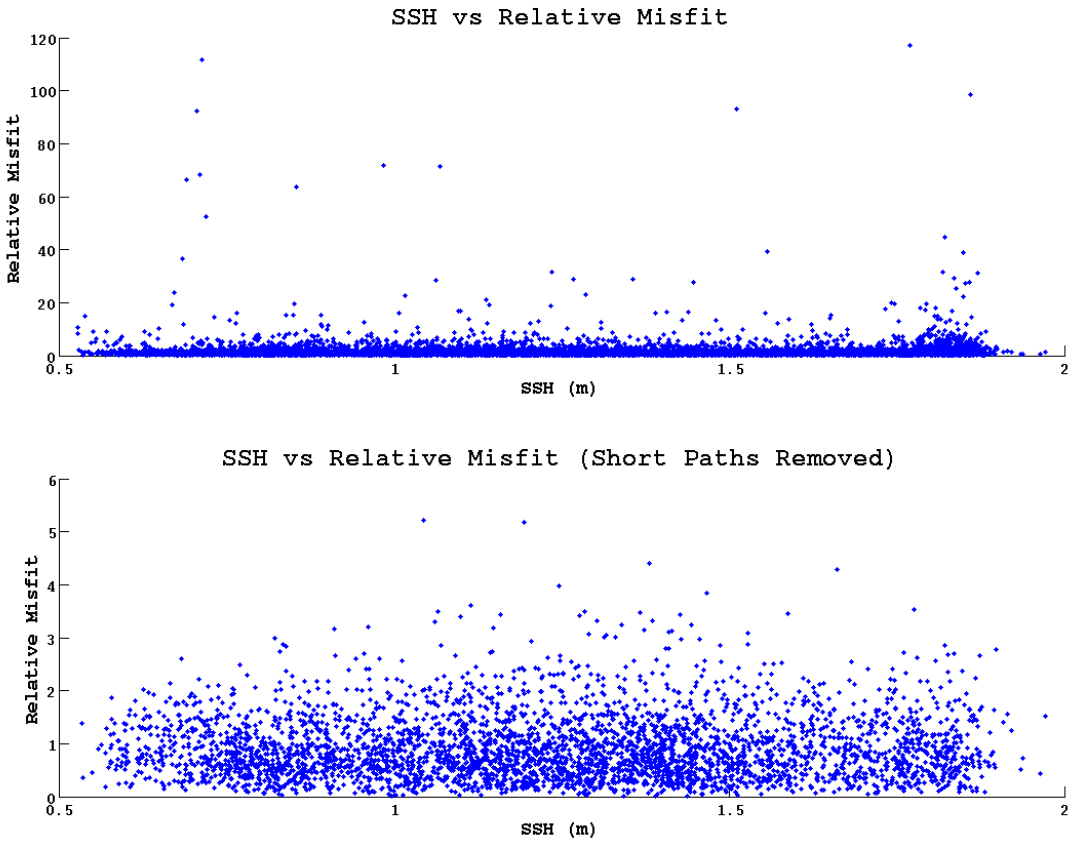


***Figure 7.21**: Scatter plots of ε_r vs SSH for all data (top) and P_{argo}>50km (bottom).*

The ratio of both small and large relative misfit cycles was then calculated for 0.03m bins based on SSH in an attempt to filter the data to discern overall long wavelength trends in float cycle distribution (Figure 7.22).

There are no significant patterns obvious in relationship between the fraction of low misfit cycles and SSH. There is greater variability in the fraction of cycles with large misfit as a function of SSH. The erratic increase in the ratio for SSH in excess of 1.75m can largely be attributed to the variability due to the limited data for regions of high SSH. The more consistent increase in the fraction of cycles with large misfit for SSH less than 0.75m may be more significant particularly as this region coincides closely with the southern branch of the Polar Front and the Southern ACC Front.

However, additional Argo data would be required to confirm that this feature is more than a statistical variation.

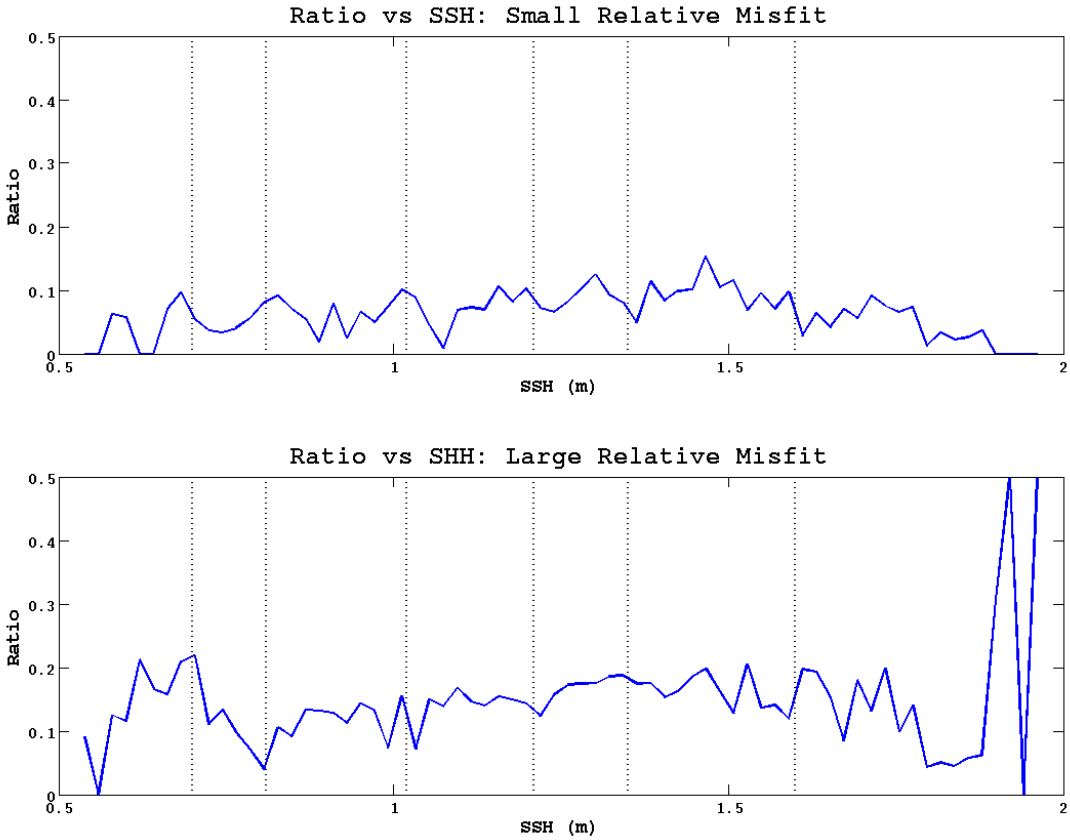


***Figure 7.22**: Ratio of the number of cycles with low ε_r to total number of cycles vs SSH at descent point (top) and ratio of the number of cycles with high ε_r to total number of cycles vs SSH at descent point (bottom). Vertical black lines indicate SSH associated with frontal features (from left to right: sACCf; southern and northern branches of the PF; southern, middle and northern branches of the SAF).*

Scatter plots of relative misfit components across and along SSH contours (not shown) were also created but again no obvious relationships or tends were apparent. The misfit data projected parallel and perpendicular to SSH contours were then sorted into bins and the fraction of cycles with high and low relative misfit components was plotted as a function of SSH (Figure 7.23).

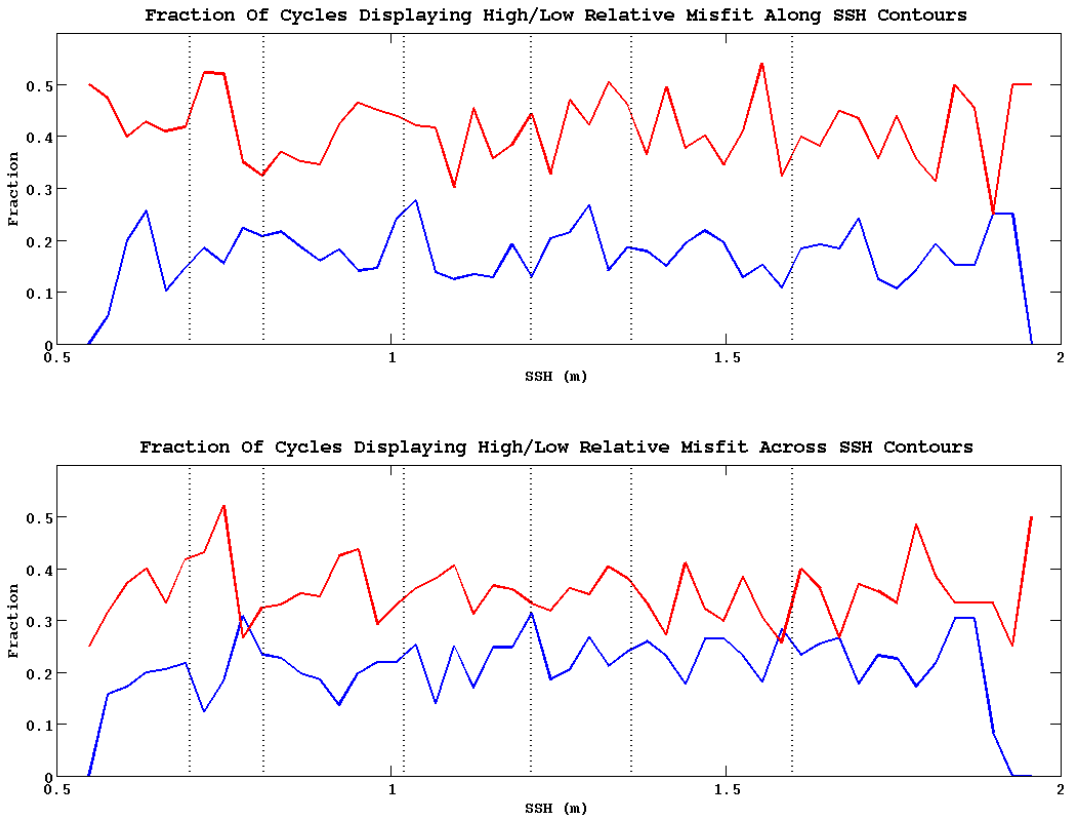


***Figure 7.23**: Fraction of cycles with high (red) and low (blue) relative misfit components along SSH contours (a) and across SSH contours (b).*

The variations in cycle ratios across and along SSH contours appear to be essentially the same. This tallies with the results obtained in section 7.3, that misfit distributions are essentially independent of orientation. For both the cycles displaying high relative misfit and cycles displaying low relative misfit neither graph shows any significant trends.

7.7 Temporal Variability

There are no obvious relationships between relative misfit and the time (Figure 7.24).

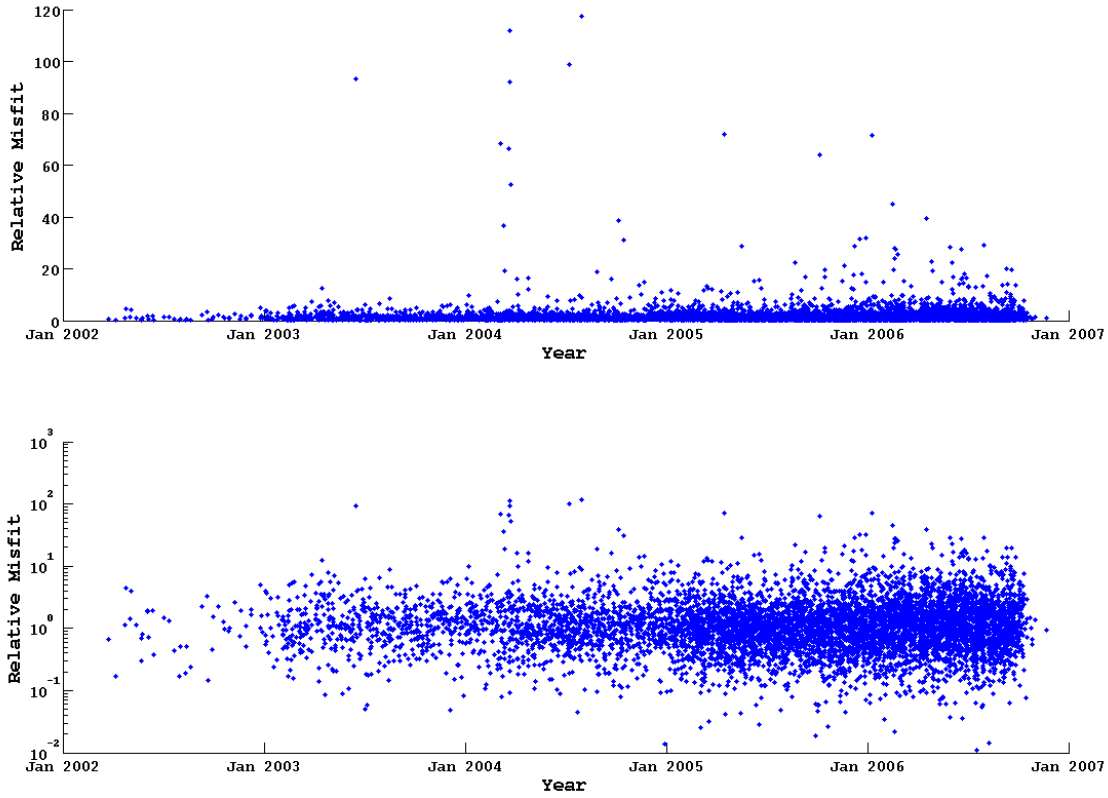


***Figure 7.24**: Relative misfit as a function of date shown on a linear axis (top) and log axis (bottom).*

In an attempt to identify underlying long-term variations in relative misfit for the two misfit populations, the fraction of cycles with high and low misfits was plotted as a function of time for three month bins from 2002 to 2007 (the range covered by both Argo data and the SatGEM fields) (Figure 7.25).

The graph of the fraction of cycles with good misfits reaches a peak at a little over 0.2 around mid-2002 before varying erratically between about 0.03 and 0.08 until mid-2004. Beyond mid-2004 the fraction of the data with low relative misfits stabilises at about 0.08. This roughly corresponds to the point when the total number of cycles per 3 month period reached 150 (Figure 7.26). This suggests the earlier variations are likely statistical in origin and a result of limited Argo data.

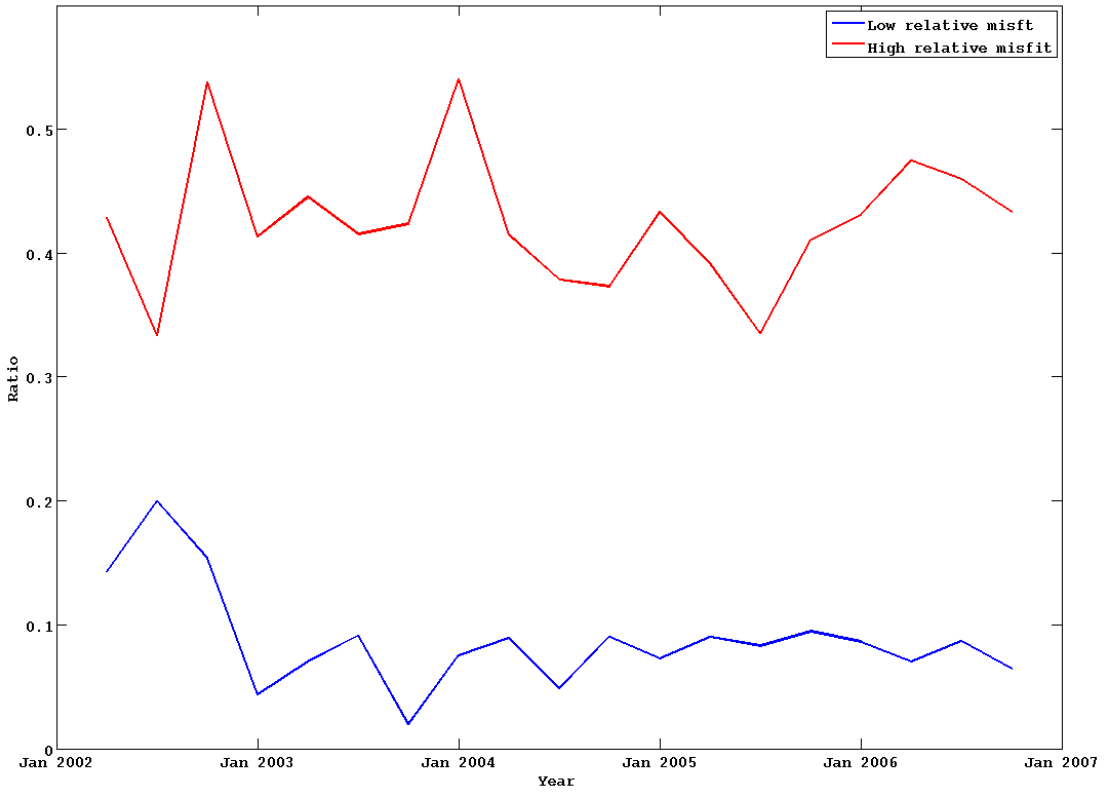


***Figure 7.25**: Fraction of data in 3 month bin with relative misfits of under 0.3 (blue) and over 1.25 (red). Restricted to cycles with P_{argo}>50km.*

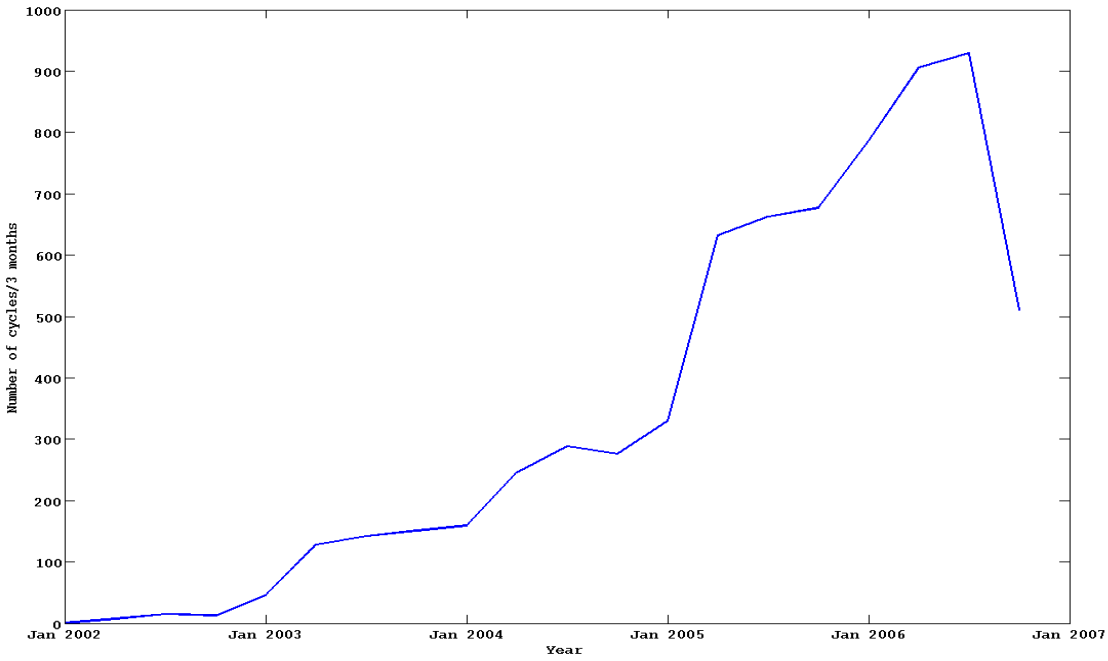


***Figure 7.26**: Argo cycles per 3 month period, 2002-2007.*

The fraction of cycles with high relative misfit shows a much more erratic pattern with variability that does not appear to decrease significantly with increasing number of floats, reflecting the fact that high misfits are not purely statistical in nature but are also related to other more chaotic factors such as the local complexity of the velocity model and the relative position of the float in the velocity field. (Figure 7.25).

7.8 Depth Relationships

Relative misfit was plotted against bathymetric depth for the float descent point on both a linear and logarithmic axis (Figure 7.27). No substantial trends are visible in either plot.

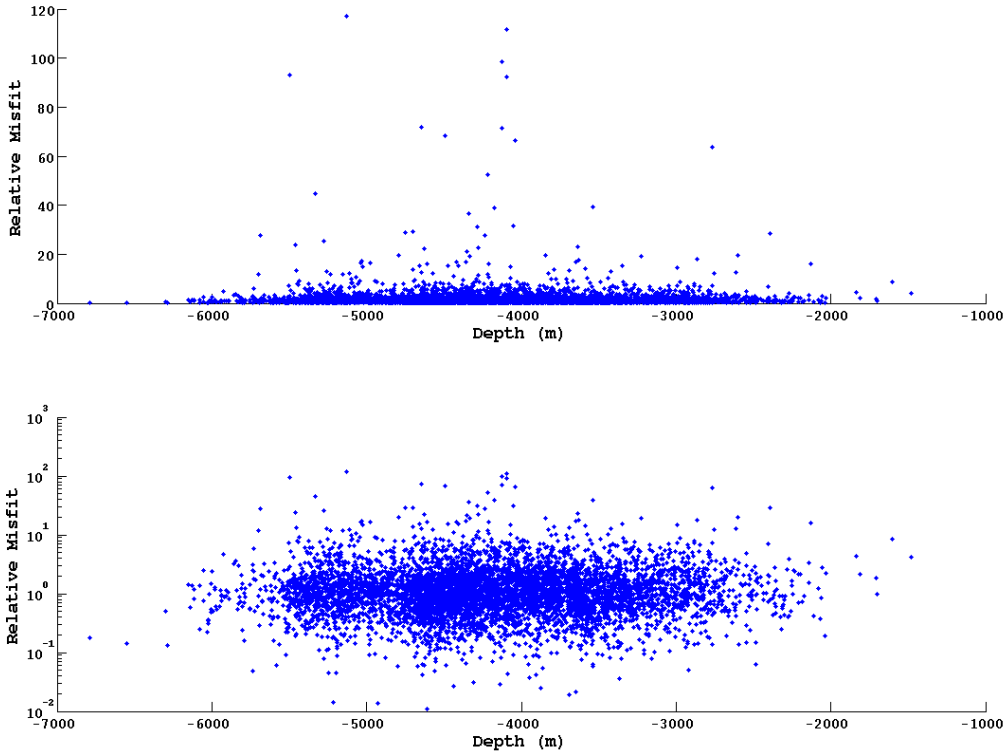


***Figure 7.27**: Scatter plots of relative misfit vs depth at descent location on a linear axis (a) and logarithmic axis (b).*

The data were then sorted into bins by depth and the fraction of cycles per bin displaying high and low relative misfits was calculated and plotted (Figure 7.28)

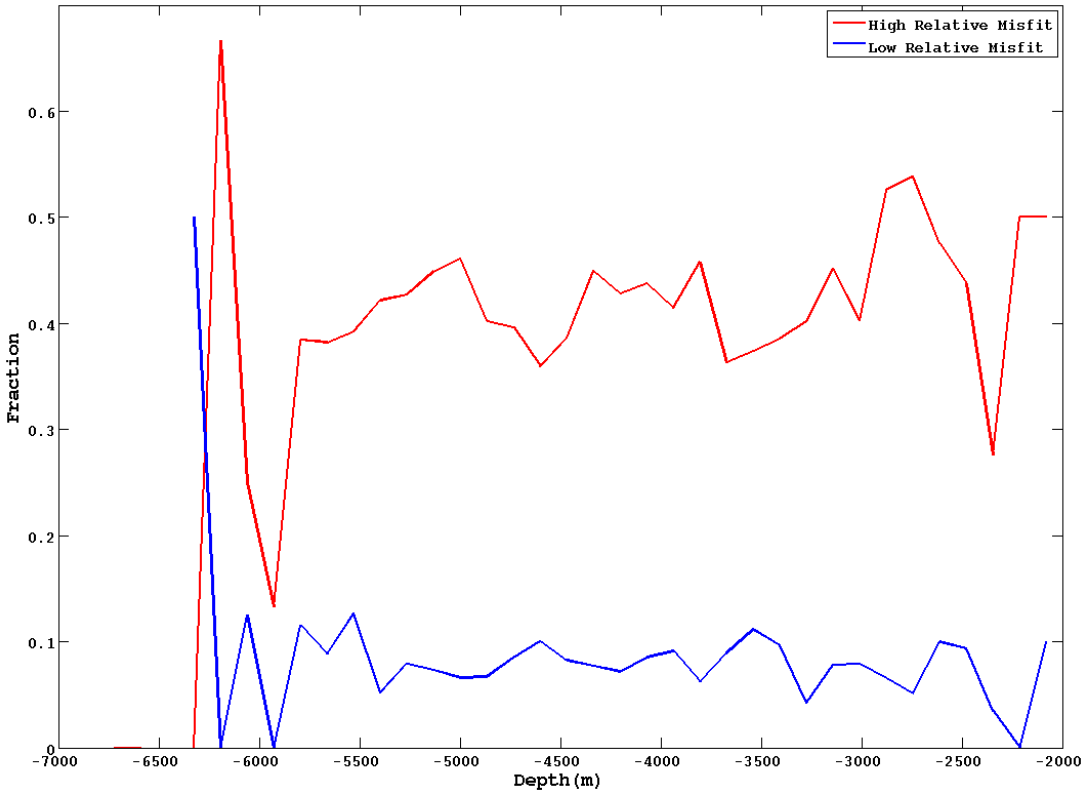


***Figure 7.28**: Fraction of cycles with high (red) and low (blue) relative misfit on a per bin basis vs depth at descent location on a linear axis.*

Data is sparse below 5500m and above 2500m. The erratic variations exhibited by both the high and low relative misfits at the extremes of the depth range are hence likely purely statistical and of no physical significance. There are no clear trends in the fraction of cycles with either small or large relative misfits as a function of depth.

7.9 Comparison with Random Data

Entire Dataset

R^2 and the correlation coefficient between simulated path length and $|P_{Argo}|$ for an entire dataset of randomly generated floats paths data are 0.0002 and -0.0131 respectively. This contrasts with the same values for the paths simulated from the SatGEM fields in which these values are 0.2780 and 0.5272 respectively. Results for both the latitudinal and longitudinal components (see Table 7.2).

	Overall	Latitudinal	Longitudinal
SatGEM R	0.53	0.46	0.53
SatGEM R^2	0.28	0.22	0.28
Random Data R	-0.01	0.004	-0.004
Random Data R^2	1.7×10^{-4}	1.9×10^{-5}	1.3×10^{-5}

***Table 7.2**: Correlation Coefficients and R^2 values for P_{sim} and P_{argo} using the SatGEM model and a synthetic dataset of random paths.*

The standard deviation (and thus variance) of the latitudinal and longitudinal distance misfits of the random data are higher than for the modelled data (85.4km compared to 63.2km for the latitudinal component and 89.0km versus 69.7km for the longitudinal component). On the basis of these simple statistical measures it would be reasonable to conclude that the SatGEM velocity fields are significantly better at predicting Argo paths than random processes.

Geographic Distribution

A circumpolar grid of R^2 values for the random paths is shown in Figure 7.29a. In the vast majority of cells the random paths explain significantly less than 10% of the variance.

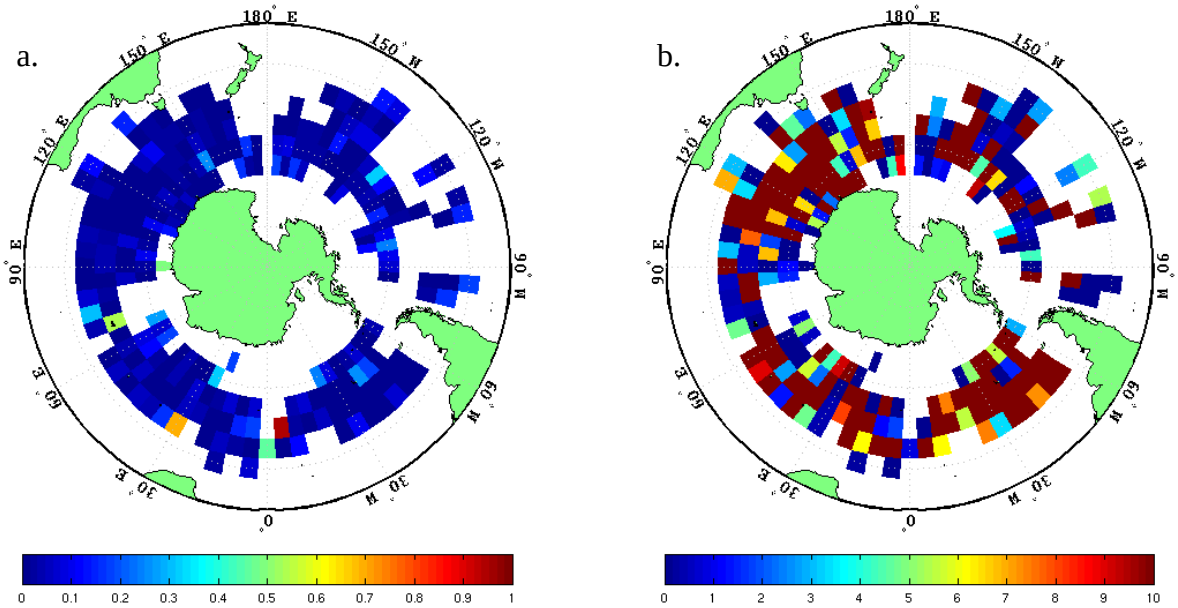


***Figure 7.29**: Circumpolar dataset of random paths R^2 values for 5° x 5° cells (a) and the normalised residuals between R^2 values for SatGEM outputs and random data (b, ratios ≥ 10 shown in dark red).*

This is reinforced by the plot of the normalised residual defined as: $(R^2_{sim} - R^2_{rnd})/R^2_{rnd}$ and shown in Figure 7.29b. Over the vast majority of the Southern Ocean this ratio is greater or equal to 10, which indicates that the residual between the R^2 values is at least an order of magnitude larger than the fraction of variance accounted for by the random paths. This indicates that the SatGEM model is a significantly better predictor of the Argo float velocity than a random velocity field.

Chapter 8: Discussion and Conclusions

8.1 The Numerical Model

Numerical modelling code to simulate the path of an Argo float through the 4D SatGEM velocity dataset of the Southern Ocean (Meijers et al. 2009a) has been successfully developed and tested. The descent, parking and profiling stages of the Argo float cycle are modelled by an incremental process that involves temporal and spatial interpolation of the velocity fields to estimate a velocity vector and hence a displacement for each time step. The code has been applied to 6984 Argo cycles from across the Southern Ocean in an attempt to validate the SatGEM velocity dataset. The model outputs the simulated Argo float trajectory and calculates parameters such as the simulated path length, the misfit vector and the relative misfit between simulated and actual surfacing positions.

For some float cycles, in relatively simple velocity fields, the relative misfit is small but for many cycles, in regions with more complex velocity structure, the relative misfit may be very large due in part to the sensitivity of the simulated path to starting position in the model and hence to the effects of small-scale structures and bifurcations in the velocity field. Variations in simulated path length relative to actual path length and data misfit were analysed from both a statistical and a spatial perspective to assess the validity of the SatGEM model.

8.2 SatGEM Validation

The distributions, means and standard deviations of the latitudinal and longitudinal components of the simulated and actual Argo path lengths are very similar. On the basis of these simple univariate statistical measures the SatGEM model reasonably approximates the large scale variability in velocity within the Southern Ocean. This conclusion is reinforced by the general similarity between the spatial distributions of actual and simulated path length although the amplitudes of the velocity anomalies vary significantly between the two datasets. On a global basis the SatGEM velocity dataset captures 28% of the variance in Argo path length. The SatGEM model produces correlations that are over an order of magnitude better than those derived between actual paths and paths generated from a purely random displacement model.

However, despite the overall statistical similarity of the simulated and actual path length data, the local misfit values are highly variable with an average relative

misfit for the entire dataset of 1.80. Relative misfit values generally decrease with increasing Argo path length and approximately 50% of the large relative misfit cycles occur for actual path lengths of less than 50km. The average relative misfit for paths greater than 50km is 0.91. The significant variability in misfit for short path lengths can in part be accounted for by the inability of the SatGEM fields to adequately represent small spatial scale or short term variations in velocity. The 1/3° spatial resolution and seven day temporal resolution of the SatGEM model mean that features of at least twice these dimensions will not be adequately sampled by the model even though they will influence actual Argo float trajectories.

The angular disparity between actual and simulated paths is normally distributed for relative misfits less than 1.25 and path lengths in excess of 50km which suggests that the source of variability in these cases may be simple statistical processes. In contrast, the uniform distribution of angular disparity for relative misfits greater than 1.25 and path lengths in excess of 50km is suggestive of more complex random chaotic perturbation processes which are likely related to complexities in the velocity field structure such as flow bifurcations.

The spatial distribution of relative misfit is quite irregular and the distribution of Argo float cycles with low relative misfit (less than 0.3) is essentially random. However, statistically significant clusters of floats with large relative misfit (>1.25) are apparent. The most prominent of these regions are areas downstream (east) of the Kerguelen Plateau and downstream of the Campbell Plateau, which are both known to be regions characterized by small-scale eddy current activity (Rintoul et al. 2001).The other area with a statistically significant proportion of large misfit cycles is in the vicinity of the Brazil-Falklands Confluence where the convergence of sub-tropical and sub-antarctic waters results in inaccuracies in GEM models derived from satellite altimetry (Sun and Watts 2001).

No significant relationships could be identified between SatGEM performance, as measured by relative misfit, and any other variables including sea surface height, frontal positions, bathymetry and time.

8.3 Further Work

This study has demonstrated by numerical forward modelling that the SatGEM velocity model provides a reasonable description of the gross 4D structure of the Southern Ocean but that it cannot account adequately for small scale flow features and

chaotic processes that perturb the local motion of Argo floats. However there are a number of additional ways that the SatGEM model could be more rigorously evaluated by alternative numerical modelling schemes.

A Monte Carlo approach, essentially utilizing the forward modelling code developed during this project, would involve initializing a number of virtual floats in the vicinity of the actual descent point and judging the performance of the model by the proportion of simulated floats that surface in the vicinity of the actual surfacing position. The spatial distribution of the virtual descent points would be decided based on a combination of positional uncertainty in the Argo float data and the resolution of the SatGEM model. The Monte Carlo approach would allow a more complete assessment of the sensitivity of model performance to initial starting position particularly in regions of complex branching flow paths.

An alternative numerical scheme would involve running two models for each float cycle, the standard forward model as described in this study, and a second model where the float path is traced in reverse from surfacing point back towards the initial descent point. This would result in two misfits for each float cycle. In this case both modelling stages would be affected by sensitivity to initial starting position in regions with complex velocity variations and a comparison of forward and reverse misfits would also provide an indirect measure of model complexity.

In addition to alternative numerical modelling schemes, there is also significant scope for more comprehensive statistical analysis of specific regions of the SatGEM model based on the simulated Argo float trajectories produced in this study, particularly if additional oceanographic data can be utilized. Priority areas for future detailed analysis would be areas of known eddy current activity, and areas with abnormally large simulated path lengths.

9. References

Belkin, IM and Gordon, AL 1996, 'Southern Ocean fronts from the Greenwich meridian to Tasmania', *Journal Of Geophysical Research*, vol. 101, pp. 3675-3696.

Carnes, MR, Teague, WJ and Mitchell, JL 1994, 'Inference of subsurface thermohaline structure from fields measurable by satellite', *Journal of Atmospheric and Oceanic Technology*, vol. 11, pp. 551-566.

Carval, T, Keeley, B, Takatsuki, Y, Yoshida, T, Loch, S, Schmid, C, Goldsmith, R, Wong, A, McCreadie, R, Thresher, A and Tran, A 2008, *Argo Data Management: User Manual version 2.1*, viewed 6th of February 2009, <http://w3.jcommops.org/FTPRoot/Argo/Doc/argo-dm-user-manual.pdf>.

de Szoeke, R and Levine, H 1981, 'The advective flux of heat by mean geostrophic motions in the Southern Ocean', *Deep Sea Research I*, vol. 28, pp. 1057-1085.

Gordon, AL 2001, 'Interocean Exchange', in G Siedler, J Church and J Gould (eds), *Ocean Circulation And Climate*, Academic Press.

Gould, J 2002, *A brief history of float developments* viewed 18 September 2008, <http://www.noc.soton.ac.uk/JRD/HYDRO/argo/history.php>.

---- 2006, *Argo, an array of profiling floats observing the ocean in realtime*, viewed 20 September 2008, <http://w3.jcommops.org/FTPRoot/Argo/Doc/Argo_new_brochure.pdf>.

Hanawa, K and Talley, LD 2001, 'Mode Waters', in G Siedler, J Church and J Gould (eds), *Ocean Circulation And Climate*, Academic Press.

Hughes, CW and Ash, ER 2001, 'Eddy forcing of the mean flow in the Southern Ocean', *Journal Of Geophysical Research*, vol. 106, pp. 2713-2722.

Lebedev, KV, Yoshinari, H, Maximenko, NA and Hacker, PW 2007, *YoMaHa'07: Velocity data assessed from trajectories of Argo floats at parking level and at the sea surface.*

Meijers, AJS, Bindoff, NL and Rintoul, SR 2009a, 'Eddy heat and freshwater transport in the Southern Ocean inferred from satellite altimetry', *To be submitted to Journal of Physical Oceanography*.

---- 2009b, 'Inferring the 4-dimensional structure of the Southern ocean with satellite altimetry', *To be submitted to Journal of Atmospheric and Oceanic Technology*.

Meinen, CS and Watts, DR 2000, 'Vertical structure and transport on a transect across the North Atlantic Current near 42°N: Time series and mean', *Journal Of Geophysical Research*, vol. 105, no. C9, pp. 21869-21891.

Mitchell, DA, Wimbush, M, Watts, DR and Teague, WJ 2004, 'The Residual GEM Technique and Its Application to the Southwestern Japan/East Sea', *Journal Of Geophysical Research*, vol. 21, pp. 1895-1909.

Nardelli, BB and Santoleri, R 2005, 'Methods for the Reconstruction of Vertical Profiles from Surface Data: Multivariate Analyses, Residual GEM and Variable Temporal Signals in the North Pacific Ocean', *Journal of Atmospheric and Oceanic Technology*, vol. 22, pp. 1762-1781.

Orsi, AH, Whitworth, T and Nowlin, WD 1995, 'On the meridional extent and the fronts of the Antarctic Circumpolar Current', *Deep Sea Research I*, vol. 42, pp. 641-673.

Park, JJ, Kim, K, King, BA and Riser, SC 2005, 'An Advanced Method to Estimate Deep Currents from Profiling Floats', *Journal of Atmospheric and Oceanic Technology*, vol. 22, pp. 1294-1304.

9. References

Pascual, A and Gomis, D 2002, 'Use of Surface Data to Estimate Geostrophic Transport', *Journal of Atmospheric and Oceanic Technology*, vol. 20, pp. 912-926.

Phillips, HE and Rintoul, SR 2000, 'Eddy Variability and Energetics from Direct Current Measurements in the Antarctic Circumpolar Current south of Australia', *Journal Of Physical Oceanography*, vol. 30, pp. 3050-3076.

Rintoul, SR, Hughes, CW and Olbers, D 2001, 'The Antarctic Circumpolar Current System', in G Siedler, J Church and J Gould (eds), *Ocean Circulation And Climate*, Academic Press.

Roemmich, D, Boebel, O, Desaubies, Y, Freeland, H, Riser, SC, Send, U, Takeuchi, K and Wijffels, S 2001, 'Argo: The Global Array of Profiling Floats', in CJ Koblinsky and NR Smith (eds), *Observing the Oceans in the 21st Century*, GODAE Project Office and Bureau of Meteorology, Melbourne.

Roemmich, D, Boebel, O, Freeland, H, King, BA, LeTraon, P, Molinari, R, Owens, WB, Riser, SC, Send, U, Takeuchi, K and Wijffels, S 1998, *On The Design and Implementation of Argo: A Global Array of Profiling Floats*, viewed 17 September 2008, <http://w3.jcommops.org/FTPRoot/Argo/Doc/Argo_Design.pdf>.

Sokolov, S and Rintoul, SR 2007, 'Multiple Jets of The Antarctic Circumpolar Current South of Australia', *Journal Of Physical Oceanography*, vol. 27, pp. 1394-1412.

Speer, K, Rintoul, SR and Sloyan, B 2000, 'The Diabatic Deacon Cell', *Journal Of Physical Oceanography*, vol. 30, pp. 3212-3222.

Stewart, RH 1997, *Introduction To Physical Oceanography*, viewed 15 September 2008, <http://oceanworld.tamu.edu/resources/ocng_textbook/contents.html>.

Sun, C and Watts, DR 2001, 'A circumpolar gravest empirical mode for the Southern Ocean hydrography', *Journal Of Geophysical Research*, vol. 106, no. C2, pp. 2833-2855.

Warren, BA and Wunsch, C (eds) 1981, *Evolution Of Physical Oceanography*, The MIT Press, Cambridge, Mass.

Watts, DR, Sun, C and Rintoul, SR 2001, 'A Two-Dimensional Gravest Empirical Mode Determined from Hydrographic Observations in the Subantarctic Front', *Journal Of Physical Oceanography*, vol. 31, pp. 2186-2209.

Wong, A, Keeley, R, Carval, T and the Argo Data Management Team 2009, *Argo Quality Control Manual: version 2.4*, viewed 16 March 2009, <http://www.coriolis.eu.org/cdc/argo/argo-quality-control-manual.pdf>.

10. Appendices: Matlab Code

Appendix 1: model_main

See description given in section 4.2.

```
vel_cor = zeros (cycles, 2);    %Velocity correction for each cycle.
Not implemented.

%Load files specifying file paths to Argo data
load /u/croach/argo_model/paths2.mat
load /u/croach/argo_model/test.mat
run_load = 1;

%Load velocity fields
if (run_load == 1)
    load /v/hpclibrary/ameijers/GEM_msla_axies.mat
    center_lon = (msla_lon(1:end-1)+msla_lon(2:end))/2;
    center_lat = (msla_lat(1:end-1)+msla_lat(2:end))/2;
    gemdate_range = datenum('14-Oct-1992')+msla_time;
    gemdate = gemdate_range(300:728);
    date_axis = find (gemdate>0);

    %Load velocity fields for 25dbar, 1000dbar and 2000dbar.
    %Modify as needed for floats with other parking and profiling
depths
    'Loading velocity fields'
    load
/v/hpclibrary/ameijers/velocities/velocities_rio_MDT_withseasons_dept
h25centerdepth_totals.mat
    load
/v/hpclibrary/ameijers/velocities/velocities_ekman_depth_25.mat
    u_surf = subvolume(centerdepthutotalrio, [1,187,300,728,1,1079])
- 100*subvolume(ekmanu, [1,187,300,728,1,1079]);
    v_surf = subvolume(centerdepthvtotalrio, [1,187,300,728,1,1079])
- 100*subvolume(ekmanv, [1,187,300,728,1,1079]);
    'Surface loaded'
    load
/v/hpclibrary/ameijers/velocities/velocities_rio_MDT_withseasons_dept
h1000centerdepth_totals.mat
    u_park = subvolume(centerdepthutotalrio, [1,187,300,728,1,1079]);
    v_park = subvolume(centerdepthvtotalrio, [1,187,300,728,1,1079]);
    'Parking depth loaded'
    load
/v/hpclibrary/ameijers/velocities/velocities_rio_MDT_withseasons_dept
h2000centerdepth_totals.mat
    u_deep = subvolume(centerdepthutotalrio, [1,187,300,728,1,1079]);
    v_deep = subvolume(centerdepthvtotalrio, [1,187,300,728,1,1079]);
    'Deep loaded'
    clear centerdepthvtotalrio centerdepthutotalrio ekmanu ekmanv
end

traj_path = Test_data;
i = 1
limit = size(traj_path);

%Enter while loop and repeat until all floats done
while(i<= numel(traj_path) && i <=limit(1))
    x = traj_path{i}
```

```
    %Pass file path, velocity fields and axes to single_float
    single_float (vel_cor, center_lon, center_lat, gemdate, cycles,
u_surf, v_surf, u_park, v_park, u_deep, v_deep, x);
    i = i+1;
end
```

Appendix 2: single_float

See description in section 4.2.

```
function [Save_loc]  = single_float (velocity_correction, center_lon,
center_lat, gemdate, cycles, u_surf, v_surf, u_park, v_park, u_deep,
v_deep, file_path)
%Load Argo float meta and trajectory files
traj_path = regexprep('/v/hpclibrary/argo_csiro/csiro_rsync/dac/path',
'path', file_path);
meta_path = regexprep(traj_path, 'traj.nc', 'meta.nc');

addpath(genpath('/usr/local/matlab-netcdf-mex'))

eval (['ncload ' traj_path]);
eval (['ncload ' meta_path]);

%Initalise timestep, max no# of cycles and index for current cycle
timestep = .25; % timestep in hours
i=1

%Load parking depth and profiling depth
zpark =PARKING_PRESSURE;
zdeep =DEEPEST_PRESSURE;

%Replace unusable entries (values of '99999') in lat, lon and cycle
%number with next usable entry
CYCLE_NUMBER = fixloc(CYCLE_NUMBER);
LATITUDE = fixloc2(LATITUDE, CYCLE_NUMBER);
LONGITUDE = fixloc2(LONGITUDE, CYCLE_NUMBER);
LATITUDE = fixloc(LATITUDE);
LONGITUDE = fixloc(LONGITUDE);
LONGITUDE = fixrange(LONGITUDE);

%Generate time arrays
td = 24*(roundn(JULD_DESCENT_END,-3) - roundn(JULD_DESCENT_START, -
3)); %descending to parking depth
tp = 24*(roundn(JULD_ASCENT_START, -3) - roundn(JULD_DESCENT_END, -
3)); %at parking depth
tprof = 24*(roundn(JULD_ASCENT_END, -3) - roundn(JULD_ASCENT_START, -
3)); %profiling
tsurf = 24*abs(JULD_DESCENT_START - JULD_START_TRANSMISSION); %on
surface

%Locate gemdates between float deployment and end of SatGEM
date_range = find(gemdate>=(JULD_START_TRANSMISSION(i) + datenum('1-
Jan-1950')));
cycle_types = numel(CYCLE_TIME); %Number of different cycle types
programmed into float

%Checks that the float was not deployed after the end of the SatGEM
data and
%that
if (numel(date_range) > 0 && cycle_types == 1)
    date_index = date_range(1) - 1;
    date_axis = find (gemdate>0); %Produces index of entries in
gemdate
```

```
    %Determine time in hours between float launch and previous
gemdate
    time = 24*(JULD_START_TRANSMISSION(i) + datenum('1-Jan-1950') -
gemdate(date_index));
    date_time_index = [date_index time]; %date_time_index provides a
convenient

    Save_lat = NaN((cycles*CYCLE_TIME(1)/timestep),1);
    Save_lon = NaN((cycles*CYCLE_TIME(1)/timestep),1);
    Save_status = NaN((cycles*CYCLE_TIME(1)/timestep),1);
    Save_phase = NaN((cycles*CYCLE_TIME(1)/timestep),1);
    Save_cycle = NaN((cycles*CYCLE_TIME(1)/timestep),1);

    loc_time_counter = 1;
    %Start of model, runs until either reaches desired number of
cycles or hits
    %end of dataset
    while (i <= (cycles -1) && date_time_index(1)+2 <= 428 && i
<=max(CYCLE_NUMBER)-1)
        x = find(CYCLE_NUMBER == i);
        z = numel(x);

        %If Argo data for the current cycle is not valid identify the
next
        %valid cycle
        while (z == 0 && i <= max(CYCLE_NUMBER))
            i = i +1;
            x = find(CYCLE_NUMBER == i);
            z = numel(x);

            if (z > 0)
                date_range = find(gemdate>(JULD_START_TRANSMISSION(i)
+ datenum('1-Jan-1950')));
                if (numel(JULD_START_TRANSMISSION) >=i &&
numel(date_range) > 0)
                    date_index = date_range(1) - 1;
                    time = 24*(JULD_START_TRANSMISSION(i) +
datenum('1-Jan-1950') - gemdate(date_index));
                else
                    date_range = find(gemdate>(JULD(x(z)) +
datenum('1-Jan-1950')));
                    if (numel(date_range) > 0)
                        date_index = date_range(1) - 1;
                        time = 24*(JULD(x(z)) + datenum('1-Jan-1950')
- gemdate(date_index));
                    else
                        date_index = date_time_index(1);
                        time = date_time_index(2);
                    end
                end
                date_time_index = [date_index time];
            end

        end

        %Load start location.
        if (i == 1 && CYCLE_NUMBER(1) ~= 0)
            curloc(1) = LAUNCH_LATITUDE;
            curloc(2) = LAUNCH_LONGITUDE;
        else
```

```
            curloc(1) = LATITUDE(x(1)-1);
            curloc(2) = LONGITUDE(x(1)-1);
        end
        curloc(3) = 0;

        %Check that location lies within latitude range and that
start date
        %plus two weeks will not run out of the SatGEM timespan
        if (curloc(1) <=-35.1522693634033 && curloc(1) >=-69.94 &&
date_time_index(1)+2 <= 428)
            u_c = velocity_correction(i, 2); %E/W velocity correction,
obselete
            v_c = velocity_correction(i, 1); %N/S velocity correction,
obselete

            Save_lat(loc_time_counter) = curloc(1);
            Save_lon(loc_time_counter) = curloc(2);
            Save_status(loc_time_counter) = curloc(3);
            Save_phase(loc_time_counter) = 1; %1 = load point, 2 =
sinking, 3 =parking 4= profiling down 5= profiling up 6= surface
drift
            Save_cycle(loc_time_counter) = i;
            loc_time_counter = loc_time_counter + 1;

            j = find(center_lat >= curloc(1));
            k = find(center_lon >= curloc(2));

            if(numel(k) == 0)
                lat_i =j(1);
                lon_i =1;
            else
                lat_i =j(1);
                lon_i =k(1);
            end

            %Find limits for reduced latitude, longitude axes
            if (abs(187 - lat_i) >= 15)
                lat_lim = 15;
            else
                lat_lim = min((187 - lat_i), lat_i);
            end

            if (lon_i >=16 && lon_i <= 1063)
                lon_lim = 15;
            else
                if (lon_i > 1063)
                    lon_lim = 1079 - lon_i;
                end

                if (lon_i < 16)
                    lon_lim = lon_i;
                end
            end

            %Construct reduced date, lat, lon axis for subvolumes
            lat= center_lat((lat_i-lat_lim):(lat_i+lat_lim));

            if (lon_i <= 1063 && lon_i >= 16)
                lon_def = center_lon(lon_i-lon_lim:lon_i+lon_lim);
```

```
            else
                if (lon_i > 1063)
                    lon_def = center_lon(1059:1079);
                end

                if (lon_i < 16)
                    lon_def = center_lon(1:20);
                end
            end
            dates = [date_time_index(1) date_time_index(1)+1
date_time_index(1)+2];

            %Take subvolumes of velocity data to improve runtime
            'Subsampling velocity fields';
            if (lon_i < 16 || lon_i > 1063)
                %Special case to be applied if
                near_edge = 1;
                u_surf2 = cat(3,subvolume(center_lat, date_axis,
center_lon, u_surf, [center_lat(lat_i - lat_lim), center_lat(lat_i +
lat_lim), date_time_index(1), date_time_index(1)+2, center_lon(1049),
center_lon(1079)]), subvolume(center_lat, date_axis, center_lon,
u_surf, [center_lat(lat_i - lat_lim), center_lat(lat_i + lat_lim),
date_time_index(1), date_time_index(1)+2, center_lon(1),
center_lon(30)]));
                v_surf2 = cat(3,subvolume(center_lat, date_axis,
center_lon, v_surf, [center_lat(lat_i - lat_lim), center_lat(lat_i +
lat_lim), date_time_index(1), date_time_index(1)+2, center_lon(1049),
center_lon(1079)]), subvolume(center_lat, date_axis, center_lon,
v_surf, [center_lat(lat_i - lat_lim), center_lat(lat_i + lat_lim),
date_time_index(1), date_time_index(1)+2, center_lon(1),
center_lon(30)]));
                u_deep2 = cat(3,subvolume(center_lat, date_axis,
center_lon, u_deep, [center_lat(lat_i - lat_lim), center_lat(lat_i +
lat_lim), date_time_index(1), date_time_index(1)+2, center_lon(1049),
center_lon(1079)]), subvolume(center_lat, date_axis, center_lon,
u_deep, [center_lat(lat_i - lat_lim), center_lat(lat_i + lat_lim),
date_time_index(1), date_time_index(1)+2, center_lon(1),
center_lon(30)]));
                v_deep2 = cat(3,subvolume(center_lat, date_axis,
center_lon, v_deep, [center_lat(lat_i - lat_lim), center_lat(lat_i +
lat_lim), date_time_index(1), date_time_index(1)+2, center_lon(1049),
center_lon(1079)]), subvolume(center_lat, date_axis, center_lon,
v_deep, [center_lat(lat_i - lat_lim), center_lat(lat_i + lat_lim),
date_time_index(1), date_time_index(1)+2, center_lon(1),
center_lon(30)]));
                u_park2 = cat(3,subvolume(center_lat, date_axis,
center_lon, u_park, [center_lat(lat_i - lat_lim), center_lat(lat_i +
lat_lim), date_time_index(1), date_time_index(1)+2, center_lon(1049),
center_lon(1079)]), subvolume(center_lat, date_axis, center_lon,
u_park, [center_lat(lat_i - lat_lim), center_lat(lat_i + lat_lim),
date_time_index(1), date_time_index(1)+2, center_lon(1),
center_lon(30)]));
                v_park2 = cat(3,subvolume(center_lat, date_axis,
center_lon, v_park, [center_lat(lat_i - lat_lim), center_lat(lat_i +
lat_lim), date_time_index(1), date_time_index(1)+2, center_lon(1049),
center_lon(1079)]), subvolume(center_lat, date_axis, center_lon,
v_park, [center_lat(lat_i - lat_lim), center_lat(lat_i + lat_lim),
date_time_index(1), date_time_index(1)+2, center_lon(1),
center_lon(30)]));
            else
                'Default case';
```

```
                near_edge = 0;
                u_surf2 = subvolume(center_lat, date_axis, center_lon,
u_surf, [center_lat(lat_i - lat_lim), center_lat(lat_i + lat_lim),
date_time_index(1), date_time_index(1)+2, center_lon(lon_i - lon_lim),
center_lon(lon_i + lon_lim)]);
                v_surf2 = subvolume(center_lat, date_axis, center_lon,
v_surf, [center_lat(lat_i - lat_lim), center_lat(lat_i + lat_lim),
date_time_index(1), date_time_index(1)+2, center_lon(lon_i - lon_lim),
center_lon(lon_i + lon_lim)]);
                u_deep2 = subvolume(center_lat, date_axis, center_lon,
u_deep, [center_lat(lat_i - lat_lim), center_lat(lat_i + lat_lim),
date_time_index(1), date_time_index(1)+2, center_lon(lon_i - lon_lim),
center_lon(lon_i + lon_lim)]);
                v_deep2 = subvolume(center_lat, date_axis, center_lon,
v_deep, [center_lat(lat_i - lat_lim), center_lat(lat_i + lat_lim),
date_time_index(1), date_time_index(1)+2, center_lon(lon_i - lon_lim),
center_lon(lon_i + lon_lim)]);
                u_park2 = subvolume(center_lat, date_axis, center_lon,
u_park, [center_lat(lat_i - lat_lim), center_lat(lat_i + lat_lim),
date_time_index(1), date_time_index(1)+2, center_lon(lon_i - lon_lim),
center_lon(lon_i + lon_lim)]);
                v_park2 = subvolume(center_lat, date_axis, center_lon,
v_park, [center_lat(lat_i - lat_lim), center_lat(lat_i + lat_lim),
date_time_index(1), date_time_index(1)+2, center_lon(lon_i - lon_lim),
center_lon(lon_i + lon_lim)]);
            end

            lon_def = gen_lon_axis(curloc, lon_def, center_lon,
near_edge);


            %Checks descent time, if td(i) implausible uses entry in
metafile
            %If both are implausible or invalid use a default
division of
            %cycle time.
            if (numel(td) >= i)
                if (td(i) <= 0 || td(i) >= 24)
                    'Unable to calculate valid time from traj file';
                    timedesc = DESCENDING_PROFILING_TIME;

                    if (DESCENDING_PROFILING_TIME == 99999)
                        'Meta file entry invalid, using "default"
time divisions';
                        timedesc = 0.04*CYCLE_TIME;
                    end

                else
                    timedesc = td(i);
                end
            else
                timedesc = DESCENDING_PROFILING_TIME;

                if (DESCENDING_PROFILING_TIME == 99999)
                    'Meta file entry invalid, using "default" time
divisions';
                    timedesc = 0.04*CYCLE_TIME;
                end
            end
            t = 0; %Temporary
```

```
            %For each timestep during descent calculate and save
            %displacement, status, phase and cycle
            while (t <= timedesc)
                lon = gen_lon_axis(curloc, lon_def, center_lon,
near_edge);
                curloc = descend(curloc, timestep, date_time_index,
lon, lat, u_surf2, v_surf2, u_park2, v_park2, dates, t, timedesc,
zpark,v_c,u_c);
                t = t + timestep;
                'sinking'
                Save_lat(loc_time_counter) = curloc(1);
                Save_lon(loc_time_counter) = curloc(2);
                Save_status(loc_time_counter) = curloc(3);
                Save_phase(loc_time_counter) = 2;
                Save_cycle(loc_time_counter) = i;
                loc_time_counter = loc_time_counter + 1;
                date_time_index = time_elapsed(timestep,
date_time_index);
            end

            %Checks time at parking depth, if td(i) implausible uses
entry in metafile
            %If both are implausible or invalid use a default
division of
            %cycle time.
            if(numel(tp) >=i)
                if (tp(i) <= 0 || tp(i) >= 480)
                    timepark = CYCLE_TIME - DESCENDING_PROFILING_TIME
- ASCENDING_PROFILING_TIME -SURFACE_TIME;
                    if (timepark <=0 || timepark >=480)
                        timepark = CYCLE_TIME * 0.8;
                    end
                else
                    timepark = tp(i);
                end
            else
                timepark = CYCLE_TIME - DESCENDING_PROFILING_TIME -
ASCENDING_PROFILING_TIME -SURFACE_TIME;
                if (timepark <=0 || timepark >=480)
                    timepark = CYCLE_TIME * 0.8;
                end
            end

            %Calculate and save displacement etc. at parking depth
            t = 0;
            while (t <= timepark)
                lon = gen_lon_axis(curloc, lon_def, center_lon,
near_edge);
                curloc = parking_depth(curloc, timestep,
date_time_index, lon, lat, u_park2, v_park2, dates, v_c, u_c);
                t = t + timestep;
                'parking depth';
                Save_lat(loc_time_counter) = curloc(1);
                Save_lon(loc_time_counter) = curloc(2);
                Save_status(loc_time_counter) = curloc(3);
                Save_phase(loc_time_counter) = 3;
                Save_cycle(loc_time_counter) = i;
                loc_time_counter = loc_time_counter + 1;
                date_time_index = time_elapsed(timestep,
date_time_index);
```

```
                [t timepark CYCLE_TIME];
            end

            %Checks profiling time, if tprof(i) implausible uses
entry in metafile
            %If both are implausible or invalid use a default
division of
            %cycle time.
            if(numel(tprof) >= i)
                if (tprof(i) <= 0 || tprof(i) >= 24)
                    timeprof = ASCENDING_PROFILING_TIME;
                    if (timeprof <=0 || timeprof >= 24)
                       timeprof = CYCLE_TIME * 0.02;
                    end
                else
                    timeprof = tprof(i);
                end
            else
                timeprof = ASCENDING_PROFILING_TIME;
                if (timeprof <=0 || timeprof >= 24)
                    timeprof = CYCLE_TIME * 0.02;
                end
            end

            %Calculate and save displacement while sinking to max
profile depth
            timeprofdesc = timeprof/3; %Assumed to be 1/3 of total
profile time
            t= 0;

            while (t <= timeprofdesc)
                lon = gen_lon_axis(curloc, lon_def, center_lon,
near_edge);
                curloc = descend(curloc, timestep, date_time_index,
lon, lat, u_park2, v_park2, u_deep2, v_deep2, dates, t, timeprofdesc,
(zpark - zdeep), v_c, u_c);
                t = t+timestep;
                'profiling down';
                Save_lat(loc_time_counter) = curloc(1);
                Save_lon(loc_time_counter) = curloc(2);
                Save_status(loc_time_counter) = curloc(3);
                Save_phase(loc_time_counter) = 4; %1 = load point, 2
= sinking, 3 =parking 4= profiling down 5= profiling up 6= surface
drift
                Save_cycle(loc_time_counter) = i;
                loc_time_counter = loc_time_counter + 1;
                date_time_index = time_elapsed(timestep,
date_time_index);
            end

            %Calculate and save displacement while ascending from max
profile depth
            timeprofup = timeprof*2/3;
            t = 0;
            while (t <= timeprofup)
                lon = gen_lon_axis(curloc, lon_def, center_lon,
near_edge);
                curloc = profile(curloc, timestep, date_time_index,
lon, lat, u_surf2, v_surf2, u_deep2, v_deep2, dates, t, timeprof,
zdeep, v_c, u_c);
```

```
                t = t + timestep;
                'profiling up';
                Save_lat(loc_time_counter) = curloc(1);
                Save_lon(loc_time_counter) = curloc(2);
                Save_status(loc_time_counter) = curloc(3);
                Save_phase(loc_time_counter) = 5; %1 = load point, 2
= sinking, 3 =parking 4= profiling down 5= profiling up 6= surface
drift
                Save_cycle(loc_time_counter) = i;
                loc_time_counter = loc_time_counter + 1;
                date_time_index = time_elapsed(timestep,
date_time_index);
            end


            if(numel(tsurf) >= i)
                if (tsurf(i) <= 0 || tsurf(i) >= 24)
                    timesurf = ASCENDING_PROFILING_TIME;
                    if (timesurf <= 0 || timesurf >= 24)
                        timesurf = CYCLE_TIME * 0.04;
                    end
                else
                    timesurf = tsurf(i);
                end
            else
                timesurf = ASCENDING_PROFILING_TIME;
                if (timesurf <= 0 || timesurf >= 24)
                    timesurf = CYCLE_TIME * 0.04;
                end
            end

            %Update date_time_index through time spent on the surface.
Now
            %obsolete.
            while (t <= timesurf)
                t = t + timestep;
                'surface drift';
                date_time_index = time_elapsed(timestep,
date_time_index);
            end
                %Save cycle details to disk
                if (cycle_types == 1)
                    file_num = regexprep(traj_path, '(\w*)/', '');
                    file_num = regexprep(file_num, '_traj.nc', '');
                    file_name = ['/u/croach/argo_model/Output/'
file_num];
                    eval(['save ' file_name ' Save_lat Save_lon
Save_status Save_phase Save_cycle']);
                end
            clear u_surf2 u_deep2 u_park2 v_surf2 v_deep2 v_park2;
        end
        i = i+1 %Increment cycle index
    end
end
clear
```

Appendix 3: descend

See description in section 4.2.

```
%descend:
%Calculates the displacement for one time step while descending.

%Inputs:
%curloc (2 element vector, containing location as lat and lon)
%timestep (scalar, containing length of time step in hours)
%date_time_index (2 element vector, containing index to most recent
%velocity field and time since last update of velocity field)
%lon, lat, time_axis (vectors containing lat, lon and time indexes of
%reduced velocity fields)
%u4d1, u4d2, v4d1, v4d2 (3d matrices containing v and u components of
%reduced velocity fields)
%time_axis
%time(scalar, time since start of descent)
%time_sink (scalar, time taken to sink)
%z (scalar, change in depth over entire descent)
%u_c (scalar, velocity correction, obsolete)
%v_c (scalar, velocity correction, obsolete)

%Outputs:
%newloc (3 element vector, containing new location as lat and lon and
'status' of cycle i.e. run into area with no data or not)

function [newloc]  = descend2(curloc, timestep, date_time_index, lon,
lat, u4d1, v4d1, u4d2, v4d2, time_axis, time, time_sink, z, v_c, u_c)

%Claculates time in weeks since start of SatGEM
size(u4d1);
numel(lat);
numel(lon);
numel(time_axis);
t = date_time_index(1) + (date_time_index(2)/168);
w = z/time_sink; %descent speed in m/hr
depth = w*time; %current depth
status = 0;

if (curloc(1) <=-35.1522693634033 && curloc(1) >=-69.94)

    %Perform initial interpolation to obtain velocities at start and
end depth
    %May add a few depths in between, but as descents are usually
quick that
    %may improve accuracy
    Vup = -interp3(lat, time_axis, lon, v4d1, curloc(1), t, curloc(2),
'linear');
    Uup = -interp3(lat, time_axis, lon, u4d1, curloc(1), t, curloc(2),
'linear');
    Vlow = -interp3(lat, time_axis, lon, v4d2, curloc(1), t,
curloc(2), 'linear');
    Ulow = -interp3(lat, time_axis, lon, u4d2, curloc(1), t,
curloc(2), 'linear');

    %Check for NaN velocities and replace with 0
    if (isnan(Vup) == 1)
        Vup = 0;
```

```
    end

    if (isnan(Uup) == 1)
        Uup = 0;
    end

    if (isnan(Vlow) == 1)
        Vlow = 0;
    end

    if (isnan(Ulow) == 1)
        Ulow = 0;
    end

    %Final interpolation between pressure level. Special clause for
if
    %pressure levels are the same, used for cycles with identical
profiling and
    %parking depths.
    if (z == 0)
        Uf = Ulow;
        Vf = Vlow;
    else
        Uf = interp1([0 z],[Uup Ulow], depth, 'linear');
        Vf = interp1([0 z],[Vup Vlow], depth, 'linear');
    end

else
    Uf = 0;
    Vf = 0;
    status = NaN;
end

%Calculate displacements in lat and lon
disp_lon=(timestep*u_c+timestep*60*60*Uf/100);
disp_lat=(timestep*v_c+timestep*60*60*Vf/100);

length_lat = 1000 * sw_dist([curloc(1) curloc(1)+1],[curloc(2)
curloc(2)], 'km');
length_lon = 1000 * sw_dist([curloc(1) curloc(1)],[curloc(2) curloc(2)
+ 1], 'km');

deltaLat = disp_lat/length_lat(1);
deltaLon = disp_lon/length_lon(1);

%Update location
newloc = [curloc(1)+deltaLat curloc(2)+deltaLon status];

%If newloc(1) shifts float location past 180 or -180 deg wrap back
round
if (newloc(2) >= 180)
    newloc(2) = (newloc(2) - 360);
end

if (newloc(2) <= -180)
    newloc(2) = (newloc(2) + 360);
end
```

Appendix 4: parking_depth

See description in section 4.2.

```
%parking_depth:
%Calculates the displacement for one time step while at parking depth.

%Inputs:
%curloc (2 element vector, containing location as lat and lon)
%timestep (scalar, containing length of time step in hours)
%date_time_index (2 element vector, containing index to most recent
%velocity field and time since last update of velocity field)
%lon, lat, time_axis (vectors containing lat, lon and time indexes of
%reduced velocity fields)
%u4d1, u4d2, v4d1, v4d2 (3d matrices containing v and u components of
%reduced velcoity fields)
%time(scalar, time since start of parking drift)
%u_c (scalar, velocity correction in m/hr, obsolete)
%v_c (scalar, velocity correction in m/hr, obsolete)

%Outputs:
%newloc (2 element vector, containing new location as lat and lon)

function [newloc]  = parking_depth(curloc, timestep, date_time_index,
lon, lat, u4d, v4d, time_axis, v_c, u_c)

%Claculates time in weeks since start of SatGEM
t = date_time_index(1) + (date_time_index(2)/168);
status = 0;

%Check that the float lies within the valid velocity range. If not
set
%velocities to 0cm/s and status to NaN
if (curloc(1) <=-35.1522693634033 && curloc(1) >=-69.94)

    %Perform interpolation
    Vf = -interp3(lat, time_axis, lon, v4d, curloc(1), t, curloc(2),
'linear');
    Uf = -interp3(lat, time_axis, lon, u4d, curloc(1), t, curloc(2),
'linear');

    %Check if velocities are NaNs, if so replace with 0
    if (isnan([Vf]) == 1)
        'V invalid';
        Vf = 0;
        status = NaN;
    end

    if (isnan([Uf]) == 1)
        'U invalid';
        Uf = 0;
        status = NaN;
    end

else
    'Out of bounds';
    Uf = 0;
    Vf = 0;
    status = NaN;
```

```
end

%Calculate displacement in degrees of lat and lon
disp_lon=(timestep*u_c+timestep*60*60*Uf/100);
disp_lat=(timestep*v_c+timestep*60*60*Vf/100);

length_lat = 1000 * sw_dist([curloc(1) curloc(1)+1],[curloc(2)
curloc(2)], 'km');
length_lon = 1000 * sw_dist([curloc(1) curloc(1)],[curloc(2) curloc(2)
+ 1], 'km');

deltaLat = disp_lat/length_lat(1);
deltaLon = disp_lon/length_lon(1);

%update location
newloc = [curloc(1)+deltaLat curloc(2)+deltaLon status];

%If newloc(1) shifts float location past 180 or -180 deg wrap back
round
if (newloc(2) >= 180)
    newloc(2) = (newloc(2) - 360);
else
    if (newloc (2) <= -180)
        newloc(2) = (newloc(2) + 360);
    end
end
```

Appendix 5: profile

See description in section 4.2.

```
%profile:
%Calculates the displacement for one time step while profiling to
surface.

%Inputs:
%curloc (2 element vector, containing location as lat and lon)
%timestep (scalar, containing length of time step in hours)
%date_time_index (2 element vector, containing index to most recent
%velocity field and time since last update of velocity field)
%lon, lat, time_axis (vectors containing lat, lon and time indexes of
%reduced velocity fields)
%u4d1, u4d2, v4d1, v4d2 (3d matrices containing v and u components of
%reduced velocity fields)
%time(scalar, time since start of profile)
%time_sink (scalar, time taken to sink)
%z (scalar, change in depth over entire descent)
%u_c (scalar, velocity correction, obsolete)
%v_c (scalar, velocity correction, obsolete)

%Outputs:
%newloc (2 element vector, containing new location as lat and lon)

function [newloc]  = profile(curloc, timestep, date_time_index, lon,
lat, u4d1, v4d1, u4d2, v4d2, time_axis, time, time_sink, z, v_c, u_c)

%Claculates time in weeks since start of SatGEM
t = date_time_index(1) + (date_time_index(2)/168);
w = z/time_sink; %descent speed in m/hr
depth = z-(w*time); %current depth
status = 0;

%Check latitude range, if invalid set velocities to 0cm/s and
if (curloc(1) <=-35.1522693634033 && curloc(1) >=-69.94)

    %Perform initial interpolation to obtain velocities at start and
end
    %depth
    Vup = -interp3(lat, time_axis, lon, v4d1, curloc(1), t, curloc(2),
'linear');
    Uup = -interp3(lat, time_axis, lon, u4d1, curloc(1), t, curloc(2),
'linear');
    Vlow = -interp3(lat, time_axis, lon, v4d2, curloc(1), t,
curloc(2), 'linear');
    Ulow = -interp3(lat, time_axis, lon, u4d2, curloc(1), t,
curloc(2), 'linear');

    %Check for NaN velocities and replace with 0
    if (isnan([Vup]) == 1)
        Vup = 0;
    end

    if (isnan([Uup]) == 1)
        Uup = 0;
    end
```

```
    if (isnan([Vlow]) == 1)
        Vlow = 0;
    end

    if (isnan([Ulow]) == 1)
        Ulow = 0;
    end

    %Final interpolation between pressure level. Special clause for
if
    %pressure levels are the same, used for cycles with identical
profiling and
    %parking depths - Legacy from adaption fom descent.
    if (z == 0)
        Uf = Ulow;
        Vf = Vlow;
    else
        Uf = interp1([0 z],[Uup Ulow], depth, 'linear');
        Vf = interp1([0 z],[Vup Vlow], depth, 'linear');
    end

else
    Uf = 0;
    Vf = 0;
    status = NaN;
end

%Calculate displacements in lat and lon
disp_lon=(timestep*u_c+timestep*60*60*Uf/100);
disp_lat=(timestep*v_c+timestep*60*60*Vf/100);

length_lat = 1000 * sw_dist([curloc(1) curloc(1)+1],[curloc(2)
curloc(2)], 'km');
length_lon = 1000 * sw_dist([curloc(1) curloc(1)],[curloc(2) curloc(2)
+ 1], 'km');

deltaLat = disp_lat/length_lat(1);
deltaLon = disp_lon/length_lon(1);

%update location
newloc = [curloc(1)+deltaLat curloc(2)+deltaLon status];

%If newloc(1) shifts float location past 180 or -180 deg wrap back
round
if (newloc(2) >= 180)
    newloc(2) = (newloc(2) - 360);
end

if (newloc (2) <= -180)
    newloc(2) = (newloc(2) + 360);
end
```

Appendix 6: time_elapsed

See description in section 4.2.

```
%Inputs:
%timestep (scalar, timestep in hours)
%date_time_index (2 element vector, containing index of most recent
%velocity field and time since last update of velocity field)

%Outputs:
%date_time_index

function [date_time_index]  = time_elapsed(timestep, date_time_index)

%Check if number of hours since last 'update' of velocity >= 168 i.e.
if more
%than a week has passed. If so increment date_time_index(1) and reset
%date_time_inex(2). If not increment date_time_index(2)
if (date_time_index(2) >= 168)
    date_time_index(1) = date_time_index(1)+1;
    date_time_index(2) = 0;
else
    date_time_index(1) = date_time_index(1);
    date_time_index(2) = date_time_index(2) + timestep;
end
```

Appendix 7: gen_lon_axis

Gen_lon_axis was implemented as a work around for the issues involving Matlab's interpolation functions not working with non-. The code does this by creating a longitudinal axis of 20° width spread either side of the 180°E meridian. Dependent on which side of this line the float is on the longitudinal axis will consider longitudes on the other side of the meridian as i.e. if a float is just to the west of the meridian gen_lon_axis will create an axis which treats 179°E as 181°W. As the functions descend, parking_depth and profile 'wrap' the longitude of a float back into the range 180°W to 180°E gen_lon_axis must be called for each time step.

```
%Inputs:
%curloc (2 element vector, containing location as lat and lon)
%def_axis (vector, default axis generated earlier)
%centerlon (vector, longitude axis)
%near_edge (logical, 0,1)
function [lon_axis]  = gen_lon_axis (curloc, def_axis, centerlon,
near_edge)

lon = def_axis;

%near_edge true?
if (near_edge == 1)

    %Approaching western limit, treat 179E as 181W, etc. for purposes
of
    %interpolation.
    if (curloc(2) < -174.5)
        lon = [flipud(centerlon(1:30)); flipud(centerlon(1049:1079))
- 360];
    end

    %Approaching eastern limit, treat 179W as 181E, etc. for purposes
of
    %interpolation.
    if (curloc(2) > 174.5)
        lon = [centerlon(1049:1079); centerlon(1:30) + 360];
    end

end

lon_axis = lon;
```